Copy No. _____

Defence Research and Development Canada
Recherche et développement pour la défense Canada

DEFENCE R&D DÉFENSE

Flash Platform Examination

Don Coady

Note: *This version does not contain active content and is for archival purposes only. To obtain an active version of this document, please contact the Defence R&D Canada – Atlantic Document Review Panel.*

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2008-291
March 2011

Canada

This page intentionally left blank.

Flash Platform Examination

Don Coady

Defence R&D Canada – Atlantic

Technical Memorandum
DRDC Atlantic TM 2008-291
March 2011

Principal Author

Original signed by Don Coady

Don Coady

Scientific Programmer

Approved by

Original signed by Francine Desharnais

Francine Desharnais

Section Head, Maritime Information and Combat Systems

Approved for release by

Original signed by Ron Kuwahara for

Calvin V. Hyatt

DRP Chair

Abstract

The Flash platform is a comprehensive infrastructure of runtime clients, developer tools, and server-side technologies for designing, developing, and delivering an extensive variety of software content and applications. Crowning this platform is the world's most ubiquitous application runtime environment - the Flash player, providing developers and end users alike the benefits of a consistent and unifying architecture across competing browsers, desktops, and mobile devices. The core strengths of the platform include multimedia, graphics, animation, and advanced text, all of which can be enhanced with sophisticated interactivity. Arguably more significant is Flash's capability related to developing and deploying second generation web-based programs with desktop style interaction, collectively known as rich internet applications.

As a research tool, Flash offers a highly versatile prototyping and rapid application development environment for exploring, refining, and experimenting with novel user interface concepts. Flash applications of varying fidelity can incorporate advanced data visualizations, novel interactivity, rich animation, and numerous forms of multimedia. Being a predominant and forefront technology of the web, Flash is ideally positioned for leveraging the exploding wealth of cloud based informational services and resources available across the internet.

This document examines the Flash platform from both an end-user and developer perspective, primarily within the context of a research tool for design, development, and experimentation involving defence and security information systems.

Résumé

La plateforme Flash est une infrastructure complète de clients d'exécution, des outils du développeur et des technologies côté serveur pour la conception, le développement et la prestation d'une vaste gamme de contenu logiciel et des applications. Cette plateforme est l'environnement d'exécution la plus utilisée dans le monde – l'application Flash player, fournit aux développeurs et aux utilisateurs finals les avantages d'une architecture unifiante et constante de tous les navigateurs, ordinateurs de bureau et dispositifs mobiles concurrents. La plateforme présente des forces de base, notamment la capacité multimédia, les graphiques, l'animation et le traitement de texte évolué, qui peuvent être améliorées à l'aide d'une capacité évoluée d'interactivité. La capacité de Flash la plus importante est sans doute sa capacité à développer et à déployer des programmes Web de seconde génération avec une interaction de type bureau, connue sous le nom d'application Internet riche (RIA).

En tant qu'outil de recherche, Flash offre un environnement très souple de développement d'application rapide et de prototypage pour explorer, améliorer et expérimenter les concepts de l'interface utilisateur novateurs. Les applications Flash de diverses fidélités peut intégrer des fonctions de visualisation des données, d'interactivité nouvelle, d'animation riche et diverses formes de multimédia. Étant une technologie du Web prédominante et de premier plan, Flash est positionné de façon idéale pour tirer parti de la richesse explosive des services et ressources informationnels de l'informatique en nuage disponibles sur l'Internet.

Ce document examine la plateforme Flash du point de vue de l'utilisateur final et du développeur, principalement dans le contexte d'un outil de recherche pour la conception, le développement et l'expérimentation impliquant les systèmes de défense et de sécurité de l'information.

This page intentionally left blank.

Executive summary

Flash Platform Examination

Don Coady; DRDC Atlantic TM 2008-291; Defence R&D Canada – Atlantic; March 2011.

Background: Research within the Maritime Information and Combat Systems (MICS) section at DRDC Atlantic has historically included a diverse set of topics relating to capturing, simulating, consuming, sharing, and analysing information critical to military operators and tactical decision makers. Many of these research projects have manifested as unique visualizations and user interface prototypes with varied levels of fidelity, often requiring development tools capable of highly customizable interactivity and rich graphical animations. Adobe Flash was used for prototyping a number of novel visualization and tactical interface concepts, and further examination of the platform's potential as a research tool was deemed beneficial.

Results: This paper presents an overview of the Flash platform and its capabilities with an emphasis on its potential as a research tool for application design, development, and experimentation in the Maritime domain as well as the broader context of Defence and Security information systems. Specific examples are presented which showcase the platform's key capabilities for graphics, animation, multimedia, and application development.

Significance: Information devices, services, applications, and networks are rapidly evolving and expanding on all fronts - accessibility, complexity, speed, capacity in addition to shear size and number. Meanwhile, the human in the loop remains steadfast at 'firmware version 1.0' indefinitely, creating ever increasing pressures for innovative visualizations and interfaces. Flash offers a highly versatile prototyping and rapid application development platform for exploring, refining, and experimenting with novel user interface concepts. Flash applications of varying fidelity can incorporate advanced data visualizations, novel interactivity, rich animation, and numerous forms of multimedia. Being a predominant and forefront technology of the web, Flash is also ideal for leveraging the exploding wealth of cloud based informational services and resources available across the internet.

Future plans: Research projects within the MICS section can vary based on many internal and external factors, however the intent is to continue leveraging Flash as a means for designing, prototyping, developing, and experimenting with future tactical visualization and user interface concepts whenever feasible. As for the future of Flash and its related technologies, having successfully dominated the PC segment, the platform is quickly being adopted across the exploding markets of other digital devices including smartphones, embedded systems, netbooks, consoles, televisions, and set-top boxes. Indeed Flash is well positioned as a write-once-run-anywhere platform for browser based applications, however, a write-once-compile-anywhere model may ultimately prove more viable for competing with the performance and device specific features of locally installed applications. Finally, looking forward, Flash is likely to remain a frontrunner in the competition among platforms for building rich internet applications through its Flex framework and related development tools.

Sommaire

Flash Platform Examination

Don Coady; DRDC Atlantic TM 2008-291; R & D pour la défense Canada – Atlantique; Mars 2011.

Historique : La recherche au sein de la section du système d'information maritime et de combat (SIMC) à RDDC Atlantique comprenait auparavant divers ensembles de sujets liés à la saisie, la simulation, la consommation, l'échange et l'analyse de l'information essentielle aux opérateurs militaires et aux décideurs tactiques. Nombre de ces projets de recherche ont résulté en des visualisations uniques et prototypes d'interface utilisateur avec divers niveaux de fidélité, nécessitant souvent des outils de développement capables d'une interactivité hautement personnalisable et des animations graphiques riches. Adobe Flash a été utilisé pour le prototypage de nombreux concepts de visualisation et d'interface tactique novateurs, et selon un examen plus poussé du potentiel de la plateforme comme outil de recherche, elle a été considérée comme bénéfique.

Résultats : Ce document présente un aperçu de la plateforme Flash et de ses capacités tout en mettant l'accent sur son potentiel comme outil de recherche pour la conception, le développement et l'expérimentation d'applications dans le domaine de la marine ainsi que dans un contexte plus large des systèmes de défense et de sécurité de l'information. Des exemples précis démontrent les capacités clés de la plateforme en matière de graphiques, d'animation, de multimédia et de développement d'applications.

Importance : Les dispositifs d'information, les services, les applications et les réseaux évoluent et prennent de l'essor rapidement à tous points de vue – l'accessibilité, la complexité, la vitesse, la capacité à accroître la taille et le nombre. Entre temps, l'humain pris dans la boucle demeure fidèle au « micrologiciel version 1.0 » indéfiniment, créant des pressions toujours croissantes pour la visualisation et les interfaces novatrices. Flash offre une plateforme très souple de prototypage et de développement d'applications rapide pour explorer, améliorer et expérimenter les concepts novateurs de l'interface utilisateurs. Les applications Flash de fidélité diverses peuvent intégrer des fonctions de visualisation des données, d'interactivité novatrice, d'animation riche et de diverses formes de multimédia. Étant une technologie du Web prédominante et de premier plan, Flash est positionné de façon idéale pour tirer parti de la richesse explosive des services et ressources informationnels de l'informatique en nuage disponibles sur l'Internet.

Plans futurs : Les projets de recherche au sein de la section SIMC peuvent varier selon de nombreux facteurs internes et externes, mais l'intention est de continuer de tirer parti de Flash comme moyen de concevoir, prototyper, développer et expérimenter les concepts tactiques futurs de visualisation et d'interface utilisateur lorsque c'est faisable. En ce qui concerne l'avenir de Flash et des technologies connexes, ayant dominé avec succès le segment de marché du PC, la plateforme est rapidement adoptée sur les marchés en pleine expansion d'autres dispositifs numériques, notamment les téléphones intelligents, les systèmes intégrés, les miniportatifs, les consoles, les téléviseurs et les boîtiers décodeurs. En effet, Flash est bien positionné comme plateforme « écrit une fois, tourne n'importe où » (WORA) pour les applications sur navigateur, mais, un modèle de type « écrit une fois, compile n'importe où » peut, au bout du compte s'avérer plus viable pour concurrencer les performances et les fonctions spécifiques au dispositif des applications installées localement. Finalement, si l'on voit plus loin, Flash a le potentiel pour demeurer en tête de course dans la concurrence entre les plateformes pour créer des applications Internet riches grâce à son cadre Flex et aux outils de développement connexes.

Table of contents

List of figures

List of tables

1 Introduction

1.1 Document Overview

1.1.1 Experiencing Flash Applications Firsthand

Much of the plethora of software and content owing to the Flash platform is highly interactive, animated, and multimedia 'rich'. Given the obvious and inherent limits of a written document, it can often be difficult (if not impossible) to provide the reader with a complete and accurate representation of the Flash-based content and applications being discussed. In other words, sometimes text descriptions and static images are simply no substitute for experiencing the 'real' thing. To partially compensate for this fact, hyperlinks to online demos, videos, and downloads have been embedded throughout the text of this document wherever relevant. Additionally, a bibliographical reference is provided for each hyperlink which includes the full URL address of the web application; readers are encouraged to try the referenced applications and demos firsthand for themselves.

Finally, in what is likely to be a first for a DRDC publication, **Annex A of this pdf document contains embedded samples of live interactive Flash content**. Readers using version 9.0 of Adobe Reader (or newer versions) can experience this content simply by navigating to the page(s) within Annex A and clicking on the embedded content. You may be prompted to allow active content to play; simply click the 'Allow' button to proceed. Like most software applications, performance will vary sharply depending on the hardware capabilities of the end user's local computer. When finished right click the content and select disable from the context menu to resume navigating the document. The purpose of including active Flash content within this document is twofold. First, it will provide another means for readers to experience specific instances of the subject matter firsthand. Secondly, and more importantly, these embedded Flash samples serve as a befitting experiment that demonstrates yet another ubiquitous aspect of the focus of this document – the Flash platform.

1.1.2 Document Layout

This document is divided into several logical sections which essentially focus on the breadth of the Flash platform including its architecture, capabilities, tools, as well as competing technologies. Following the Introduction, Section 2 defines and describes the Flash platform runtime clients and server components, highlighting their capabilities with specific examples. The subsequent Section 3 presents a selection of extended Flash capabilities, most of which are loosely related to research themes within the MICS section such as collaboration, multi-touch, visualization, human-computer interfaces, and social networking. Development tools for the Flash platform are presented in Section 4, organized into separate groupings for Adobe, open source, and third-party providers. The extendibility of Flash tools using application programming interfaces (APIs) and components is discussed in Section 5. Competition and alternatives to the Flash platform are explored briefly in Section 6. Some of the more common criticisms relating to Flash are outlined in Section 7. A consolidated reference list of online Flash resources the author has found useful are listed in Section 8, and further subdivided into subsections such as

Learning, Events, Communities, Blogs, and APIs. The near future of Flash and related technologies is considered in Section 9, while the final Section 10 concludes with a summary of the Flash platform and its potential, particularly with regard to being a research tool. As mentioned, Annex A provides several embedded samples of Flash content which demonstrate a few of the graphics, animation, and interactive capabilities of the platform.

1.2 A Brief History

Flash is a technology born out of, and intimately tied to, the architecture of the World Wide Web. Indeed for the most part, the Flash platform has progressed in concert with the evolution of information and content on the web. Early internet content consisted largely of static text and thumbnail images (e.g., jpeg, gif, and png) embedded within relatively anorexic looking web pages using hyper-text mark-up language (html). Not surprisingly then, text and images in these early formats are core to the html standard and can be rendered *natively* by any web browser – no plug-ins required.

Over time the popularity of the web and the volume of online content exploded. What began as a handful of researchers and a few skimpy html documents has grown to over a billion people and massive repositories of information, media, software, and services spread across a complex and sprawling global system of interconnected networks. It is this very growth which has necessitated the appearance and subsequent popularity of search engines as well as user-driven tagging, voting, and rating schemes. The variety of information forms available on the web has evolved dramatically, fuelled largely by the rapid progression of computing hardware and steadily increasing network bandwidths. Simple html documents have been gradually supplanted by a myriad of animation (e.g., ads), interactivity (e.g., games), multimedia (e.g., audio and video), and web applications.

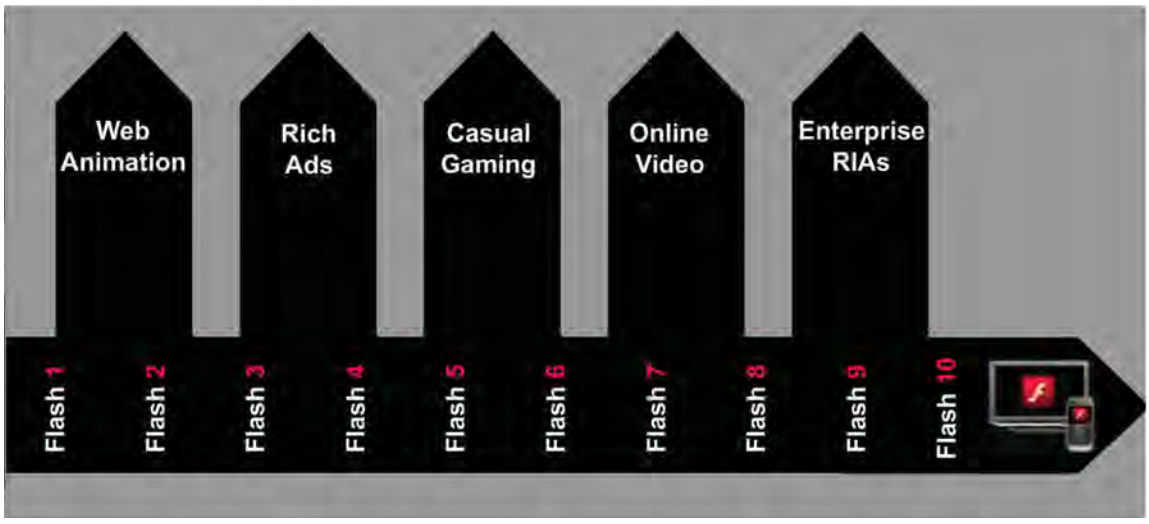

Figure 1: The Flash player has evolved lock-step with the internet to enable many forms of online content and applications.(Image modified from [1])

As the ‘doorway’ to the internet, the lowly web browser with its page-centric paradigm has never been able to keep pace nor directly address the rapidly growing content demands of the web which extend well beyond the duties of rendering simple html web pages. Thus was born the era of web browser plug-ins: optionally installable add-ons which enable non-native html content to be rendered inside the browser.

Early-on the landscape of web browser plug-ins for handling non-native content was quite fragmented. For example, to watch video web surfers might have needed Apple QuickTime for a website, Real Networks RealPlayer for another, and Microsoft Windows Media Player for yet another. Likewise, other specific file formats for images, audio, and animation unsupported by the browser had a similar fragmentation of plug-ins. Fortunately for web developers and end users alike, Adobe’s Flash player emerged as a predominant standard among plug-ins which supports most of the popular types of media content common on the web today. However the Flash dominance over non-native browser content does not extend equally into the battle for a next generation web application platform. To date the competition for a rich internet application (RIA) platform capable of delivering web applications which look and feel like desktop installed programs is very tight between competing solutions such as java, Silverlight, Flash, and html 5.

1.3 The Web and Software Architecture

In addition to expanding the shear quantity and complexity of data, the internet continues to be a primary driver in the overall evolution of modern information systems; redefining how information is created, disseminated, and accessed. The impact of internet derived technologies such as html and Flash is not limited to ‘online’ devices, but rather extends equally into stand-alone, local, corporate, and occasionally-connected environments. From this perspective, Flash and web based technologies in general, are by no means exclusive to the internet. For example, the term ‘web application’ essentially refers to any application delivered inside a web browser. In a typical scenario, a user’s web browser retrieves the application from a web server located on the internet. However, this is certainly not the only deployment model for a web application. The application could reside on a corporate intranet server, a local area network server, or even on the user’s local machine. The ‘web’ in ‘web application’ simply means the application runs inside a web browser, and it may or may not have any connection or association with the internet.

This same logic extends equally to any of the numerous architectural technologies commonly associated with the internet. Cloud based computing essentially involves server-side processing; the servers could be local, corporate, or internet based. Software as a Service or ‘SaaS’ is another ‘web’ term for web applications hosted on an internet-based server and executed completely in the browser, but again the client and server can be connected via any type of network. Service-oriented architecture (SOA) refers to a popular software development style in which businesses, governments, and other organizations expose (publish) their services so they can easily be incorporated (consumed) during application development and deployment. Again, while it is often synonymous with the web, SOA can be applied in the development of any client-server software, regardless of the network environment. Thus the architecture of the web is by no means unique or limited to the internet. Indeed quite the opposite is true; the architectural trends of the web heavily influence or more often define software development in general, proliferating across all manner of non-internet based applications, networks, and devices.

While the software architecture of the web may not be unique, the same cannot be said for much of the content, resources, and information services available from the internet today. The breadth of content and services unique to the web is extensive, and includes massive repositories of community-driven content on websites such as youtube, flickr, twitter, Wikipedia, and facebook. Other web exclusive services include search, mapping, weather, traffic, news, and countless other information feeds. As a predominant and forefront technology of the web, Flash is ideally positioned to leverage the wealth of services and content unique to the web.

2 The Flash Platform

2.1 General

For many internet users, 'Flash' is simply a plug-in for their web browser without which many websites will not work properly. This plug-in is Adobe's Flash player; it is in fact the cornerstone of the Flash platform. However, in terms of the overall Flash platform, the player is the proverbial 'tip of the iceberg'. In reality, the Flash Platform includes numerous interrelated and supporting technologies for the design, development, and delivery of Flash content and applications.

The Flash 'ecosystem' includes an ever expanding selection of tools, frameworks, components, APIs, languages, and client-server technologies. Among these, Flex is a particularly notable and extensive framework used to develop full-featured data driven web applications with sophisticated user interfaces. Unlike the Flash player itself, a considerable portion of Flash's supporting infrastructure is available through non-Adobe sources including third-party vendors, open source organizations, and individual developers. That said, Adobe does provide two of the more popular and recognizable design and development tools for Flash, namely Flash Professional and Flash Builder (previously known as Flex Builder[1]).

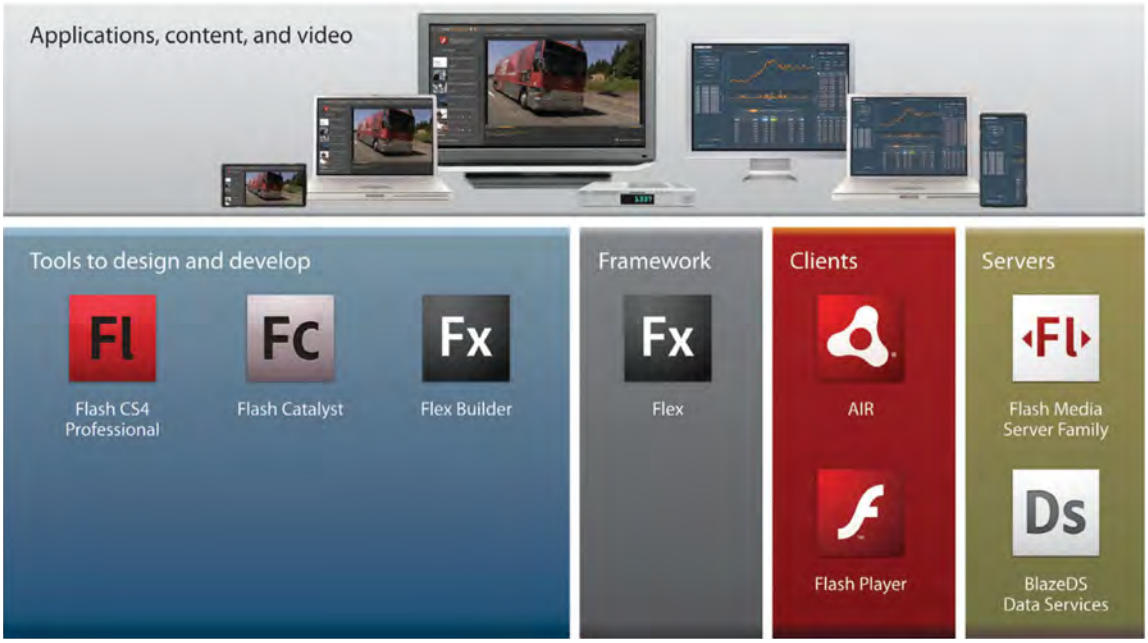

Figure 2: Adobe's Flash player is the cornerstone of the Flash platform and is supported by a myriad of tools. The Flash platform is completely cross-platform and device agnostic spanning gaming consoles, set-top boxes, netbooks, desktops, and smartphones[4].

[1] Adobe renamed their 'Flex Builder' developer tool to 'Flash Builder' coinciding with the release of version 4. The two names are used synonymously throughout this paper.

Of course, the Flash player and its supporting tools would be nothing without content and applications to deliver. The diversity of content which can be rendered using the Flash player is extensive and includes audio, video, 2D and 3D graphics, images, animations, and advanced text. In addition, the Flash player is an application runtime environment; a platform for running all manner of web applications. Flash applications range from expressive theme-oriented design creations in advertising, games, and websites to highly pragmatic data-driven business applications featuring sophisticated user interfaces, advanced visualizations, rich interactivity, and versatile connectivity. More recently, Flash has emerged as a frontrunner platform for building and delivering Rich Internet Applications (RIAs): web applications which mimic the look and feel of installed desktop applications.

Figure 3: The Flash player enables many forms of web content and applications. (Image from [2])

Capabilities and content are meaningless however if nobody has or uses the runtime, and this is arguably the strongest case for Flash, both as a media delivery and application runtime environment. Flash enjoys an enviable installed user base of over 98% of the world's computers connected to the internet[3]. Adobe further extends the Flash player reach outside the browser to the desktop using the Adobe Integrated Runtime (AIR), much in the same manner as the Java Runtime Engine (JRE). Adobe has also been busy pushing Flash technology beyond the client side and into the server space with scalable, performance-oriented products for serving video and data (e.g., Flash Media Server and BlazeDS Data Services). Finally, the Flash player is increasingly installed on non-PC hardware platforms like gaming consoles, embedded systems, digital appliances, and last but certainly not least, mobile phones.

2.2 Flash Capabilities

This section introduces some of the core capabilities of the Flash platform. More specifically, graphics, animation, video, text, and application development features are discussed. Both references and hyperlinks to 'live' samples are also included within each subsection.

2.2.1 Adobe Sampler Applications

Adobe provides a few sampler web applications on their website for experiencing several capabilities of the Flash player firsthand, two of which are particularly noteworthy. Firstly, the 'Double Identity'[5] web demo is a quick and easy way to sample some of Flash player's video, text, graphic, and animation capabilities. Secondly, the 'Tour de Flex'[6] application provides an extensive sample set of Flash capabilities relating to application development. Tour de Flex includes working samples of user interface (UI) controls, maps, data visualizations, cloud services, and data connectivity. For developers, the sampler application also includes a panel for viewing the code used to implement the feature(s).

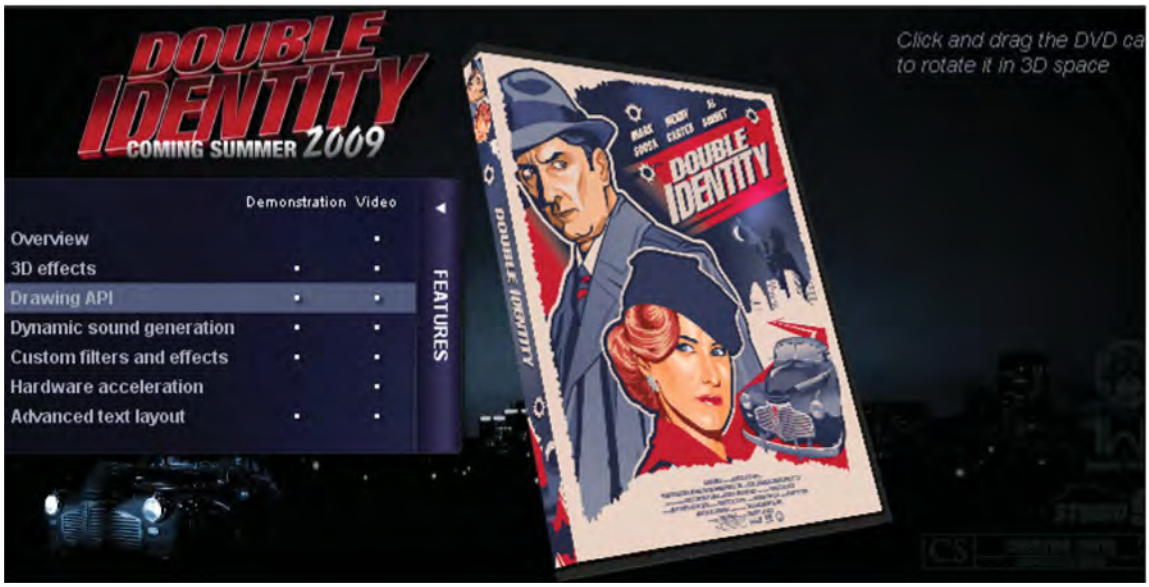

Figure 4: The Adobe Flash Player 10 demo application 'Double Identity'[5] provides a good sample of some of the Flash player's built-in capabilities including interactivity , drawing, text layout, sound, 3D, and image effects.

Figure 5: The Tour de Flex[6] demo provides dozens of interactive examples (code included) which showcase Flash's application development capabilities.

2.2.2 Graphics

Graphics within Flash are typically vector based, although image formats like bitmaps are supported and manipulable as well. Vector graphics are represented very efficiently using mathematical formulas, whereas in a bitmap each individual pixel must be represented separately. Consequently vector graphics require significantly less memory, storage, and download bandwidth. Additionally, vector graphics can be modified (e.g., enlarged) with no loss in quality. Not surprisingly, many design-time GUI (Graphical User Interface) based drawing tools (e.g., brush, pen, line, rectangle, and bucket), as well as runtime drawing APIs, are vector based.

Creating graphics for Flash can generally be accomplished in any combination of three methods: design tools, import and creation with built-in APIs. One of the most popular practices is to simply use 'design-time' GUI based drawing tools common within Flash design-oriented tools. For example, Adobe's premier Flash design tool - Flash Professional, includes several such tools accessible using its drawing toolbar (Figure 85).

Importing graphics created using other graphic design tools is another popular option for Flash designers. The graphic file formats which can be imported will depend largely on the Flash design tool being used. Adobe's suite of design tools is particularly strong in this regard, as it offers a high degree of integration and compatibility for sharing visual assets (Figure 6).

Figure 6: Flash Professional's supported import formats include several graphic and image formats.

A third option is to create (e.g., draw) visual assets at runtime using Flash drawing APIs. This programmatic approach to creating graphic objects is usually reserved for developers due to its code-centric nature. While using code is an alternate technique for drawing and animation, it is a necessity for enabling interactivity.

In practice it is not uncommon to combine drawing techniques, for example creating a complex graphic using GUI drawing tools during design-time and later enabling and modifying it using a drawing API at runtime.

Regardless of how Flash graphics are created, the Flash player is easily capable of rendering 2D graphics of all shapes, sizes, and colors. Flash also supports an impressive array of effects which can be applied at design-time or runtime including bevels, transparency, drop shadows, and glow filters. Thus creativity is really only limited by the designer's imagination and the Flash design tool being utilized. Performance limits will depend largely on the end user's machine, but as a general rule the Flash player is pretty efficient with basic 2D vector graphics. Overuse of special effects like drop shadows, poorly written code, large hi-resolution images, or simply too many displays objects will progressively kill rendering performance. Generally speaking, rendering 3D graphics in Flash often lowers performance. GPU hardware acceleration of vector graphics was introduced in version 9 of the Flash player, however support for this feature is device specific.

Flash has long had support for masking – the ability to use any shape or graphic as a visibility filter for content (e.g., graphics, images, and video) 'beneath' or 'behind' it. A mask graphic is no different than any other graphic in Flash, other than it is designated a mask object at runtime, or at design-time by being added to a mask layer.

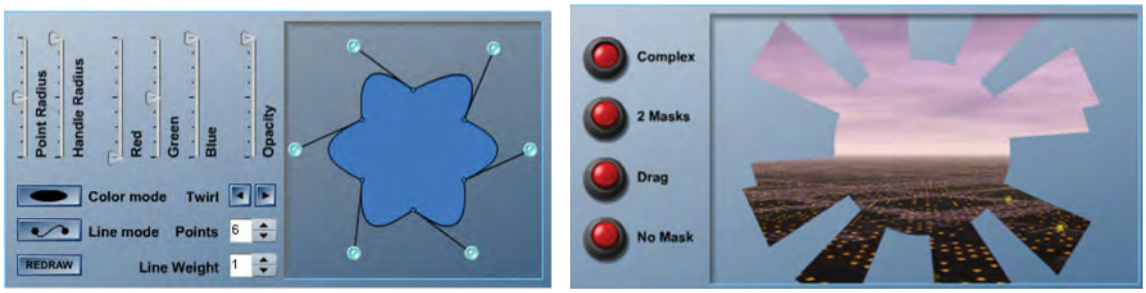

Figure 7: The Flash player's drawing API[7-8] supports all manner of 2D vector graphics, effects, and masking. Annex A contains several interactive samples which demonstrate some of Flash's graphics capabilities.

Collision detection is another handy feature within Flash which tests during runtime whether or not two separate graphic objects overlap in 2D space. For example, in a maze style game, collision detection prevents the 'hero' from walking through maze walls. Within user interfaces, collision detection can be used in many direct object manipulation interactions, like dynamically checking the target or drop area during drag-n-drop operations (see the Logicly[9] demo in Annex A for an example). Interactive examples of Flash collision detection can be found in Sections 0, A.1.6, A.1.14, and 0 of Annex A.

Coinciding with version 10 of the Flash player, Flash Professional CS4 introduced native support for 3D drawing and animation *effects*. Although all graphic objects are still planar, they can now be manipulated in three dimensions, with the addition of the z-axis. This includes translation and rotation of (flat) objects about any of the 3 axes, the so called "postcards in space". The rendering quality and mouse responsiveness for 2D graphics being manipulated using Flash professional's 3D translation and rotation tools have however been criticized as relatively poor[10].

While Adobe offers no tools for creating true 3D graphics in Flash, there are no shortage of third-party options. Numerous 3D tools and APIs are available from commercial and open-source organizations. For more information and examples of Flash's 3D graphics capabilities see Section 3.8.

As mentioned, image use in Flash tends to be less favoured over vector graphics, but nonetheless well supported for both rendering and manipulation. The Flash player has long had support for all the common image file formats including gif, png, bmp, and jpeg. Images can either be embedded in the Flash swf file or loaded at runtime. The Flash player is also one of a handful browser plug-ins capable of providing panoramic image viewing experiences[11], popular in tourism and realty applications. Users can zoom and pan 360 degrees within an image for a more immersive experience. Google Street View[12] is an interesting and familiar Flash application which merges image panoramas with map navigation. Using Google Street View, users can

virtually visit and traverse select map locations with a first person perspective as if they were physically onsite.

Figure 8: Google Maps Street View[12] takes advantage of Flash's support for full 360° interactive panoramic image viewing.

Adobe's Pixel Bender[13] is a popular and free tool designers can use to easily create their own Flash image filters and effects (e.g., crystallize, vortex, spherize). Pixel bender source files are written and saved in a form of xml called Pixel Bender Kernel (PBK) and compiled into Pixel Bender JIT (PBJ) binary files which in turn are embedded or loaded at runtime and applied by the Flash player. It is important to note most of the image filters and effects possible in Flash using Pixel Bender apply equally to vector graphics and video. Adobe hosts a Pixel Bender Exchange[14] area on their website where designers and end users can share their custom Pixel Bender files (Figure 9).

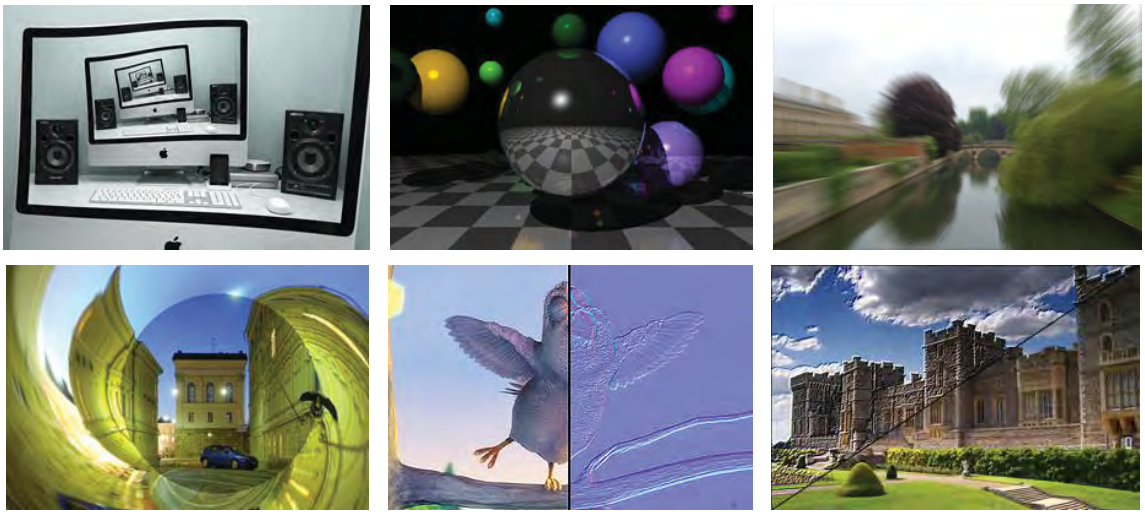

Figure 9: Examples of Pixel Bender effects available on Adobe Exchange[14]. Top from left: Escher's Droste Effect, Raytracer, Zoom Blur. Bottom from left: TubeView, Smart NormalMap, Sharpen.

Antti Kupila's Pixel Bender levels example[15] is a demo for experimenting with Pixel Bender's image filtering capabilities (Figure 10). The color and luminosity levels of an image can be interactively adjusted using each channel's histogram.

Figure 10: Real-time image manipulation is demonstrated using the Pixel Bender levels demo[15] .

Finally, it is interesting to note that Pixel Bender can be used for purposes other than image manipulation. For example, Pixel Bender can be used to speed up complex mathematical calculations in Flash applications. Not only is Pixel Bender optimized for many specific math functions, even more importantly it runs in a separate thread than the Flash player. The list of math functions supported by the Pixel Bender Toolkit includes:

- *sin(x) - Trigonometric sine function*
- *cos(x) - Cosine function*
- *tan(x) – Tangent function*
- *asin(x) - Arcsine (inverse sine) function*
- *acos(x) - inverse cosine (arccosine) function*
- *atan(x) - Arctangent (inverse tangent) function*

- *atan(x, y) - Arctangent (inverse tangent) function*
- *exp(x) - exponential function*
- *log(x) – Logarithm function*
- *pow(x, y) - power of function*
- *reciprocal(x) - multiplicative inverse function.*
- *sqrt(x) - square root function[16].*

Any CPU heavy applications which use these functions could potentially benefit, including:

- *Compression algorithms*
- *3D calculations* (e.g., 3D graphics APIs)
- *Surveying, navigation and astronomy position calculations*
- Wave additions for producing audio signals
- *Ballistic trajectories*
- *Space flight and satellite positioning*
- *GPS and cellphones* (algorithms) *rely*ing *on triangulation and formulas involving sin/cos*
- *Signal transmission, such as TV or radio broadcasting, involv*ing *waves described with sin/cos waves[16].*

2.2.3 Audio

Audio is of course an integral part of many software applications, games, videos, and websites. Many application interfaces enhance user interactions using sound effects – the audible 'click' of a virtual button, the dull synthetic dong of a system halt error in windows, and numerous other audio alerts generated by everyday applications. Of course video and its accompanying audio is increasingly common thanks in part to larger network bandwidths and client device capabilities. Text-to-speech and voice recognition systems are other niche areas where audio is vital. Acoustic recording, playback, and analysis are also essential to many scientific fields, particularly within the subsurface domain. Finally audio in the form of voice-over-IP (VOIP) communications can increasingly be found in real-time collaboration applications.

The Flash player supports the majority of the popular audio formats in use today including ADPCM, HE-AAC, MP3, FLV (audio), Nellymoser, and Speex (a hi-fidelity open source codec for voice). A more detailed overview of the Flash player's audio capabilities[17] is available on the Flash Developer area of the Adobe web site.
Support is not limited to audio playback; developers can use Flash technology to dynamically generate, modify, stream, and record sound. For example the Flash player can access and retrieve audio from a user's microphone (or webcam) and broadcast or stream the data in real-time.

Sonoflash[18] and Tenoran[19] are examples of web applications that allow users to interactively generate their own sounds and music respectively. You can use the Sonoflash Player's interface control to create a desired sound effect and then click the 'Generate Code' button to automatically create the Flash programming code to recreate the sound at runtime. Sonoflash's ActionScript-based audio components can be used to generate real time fully interactive sounds, no need to embed or load mp3 or wav files.

The “Tenoran” music generator allows budding composers to exercise their acoustic creativity. “The 16 x 16 on-off button grid is clickable. Drum sounds are assigned to the bottom 5 rows and the remaining rows are for bass sounds. Toggle sounds on and off by clicking the buttons and they will play as the vertical line passes over the buttons. Controls along the bottom allow you to set the tempo and sound effects.” [19]

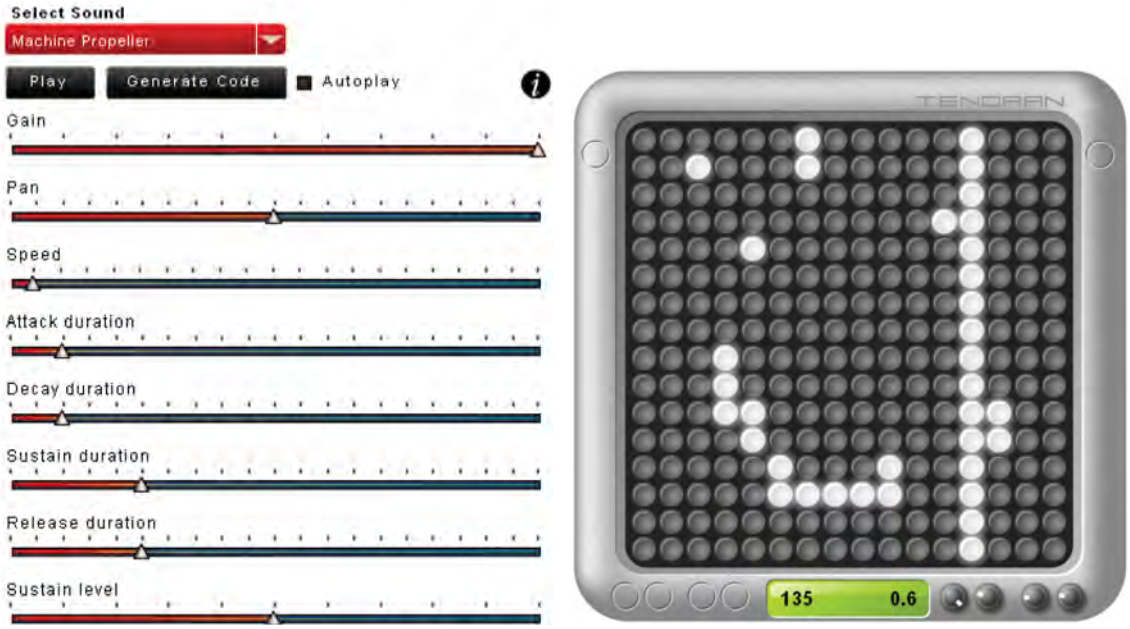

Figure 11: Users can interact with the Sonoflash[18] (left) and Tenoran[19] (right) web based players to generate sound dynamically.

Noteflight[20] is an interesting browser based application that musicians can use to compose and listen to new audio masterpieces. The application features a rich interactive GUI and familiar music notation. Users can save and share their compositions via the noteflight website.

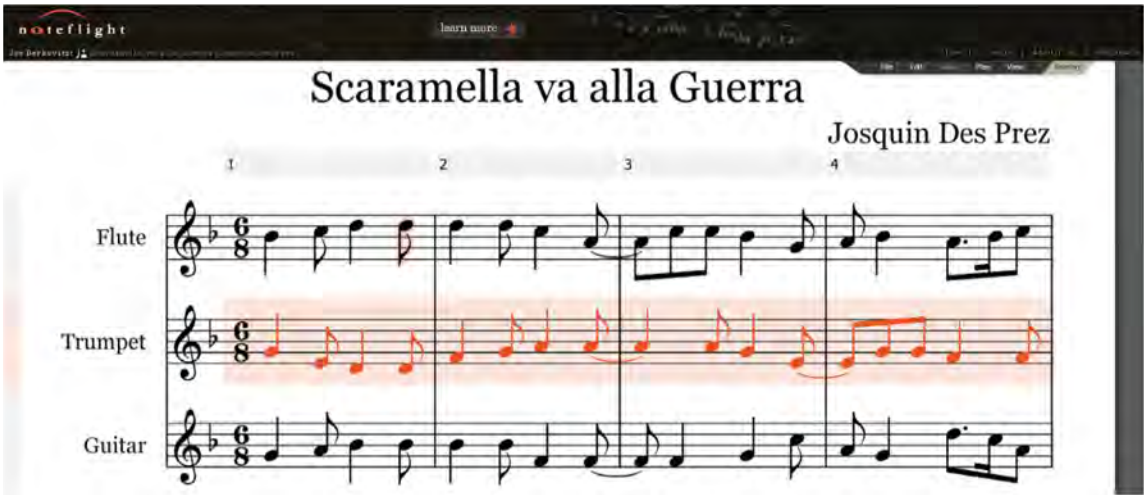

Figure 12: “Noteflight is an online music writing application that lets you create, view, print and hear music notation with professional quality, right in your web browser.”[20]

Other examples of Flash audio applications are capable of reading or capturing sound files and parsing them into distinct channels which can be separately altered and remixed as desired. Two good examples of such tools are the Audiotool[21] by Hobnox and the Myna Audio Editor[22] by Aviary.

Figure 13: The Audiotool[21] by Hobnox offers an array of virtual audiophile equipment which can be wired together and tweaked ad infinitum using the tool's GUI.

Audiotool's graphical interface provides an extensive selection of virtual audiophile equipment which can be wired together and tweaked ad infinitum. The possibilities for composing and mixing tracks are limitless; basically Audiotool is an entire audio recording studio within your web browser.

The Myna Audio Editor eschews the equipment paradigm for a more traditional track based GUI. Users can upload, share, mix, and record audio tracks using timeline based visualizations.

Figure 14: The Myna Audio Editor is a powerful web based tool to "remix music tracks and audio clips. Apply sound effects and record your own voice or instruments"[22].

2.2.4 Video

Video is another core strength of the Flash platform, including capture, streaming, and playback. On the tooling side, Adobe bundles built-in video playback components with both Flex Builder (Adobe's developer IDE) and Flash Professional. There is also a wide selection of third-party video components available on the web via commercial and open source licenses (see Sections 5.1.1 and 5.1.2 for more info on components). You can of course build your own video player completely from scratch using Flash's built-in ActionScript programming language, or leverage Adobe's open-source media framework (OSMF).

Flash has emerged as the dominant standard for digital video on the internet; over 80% of video on the web is Flash based [1]. Not surprisingly, the Flash video format (FLV) is also the most common video format, although the player supports other popular formats including MP4, QuickTime MOV, and high-definition H.264.

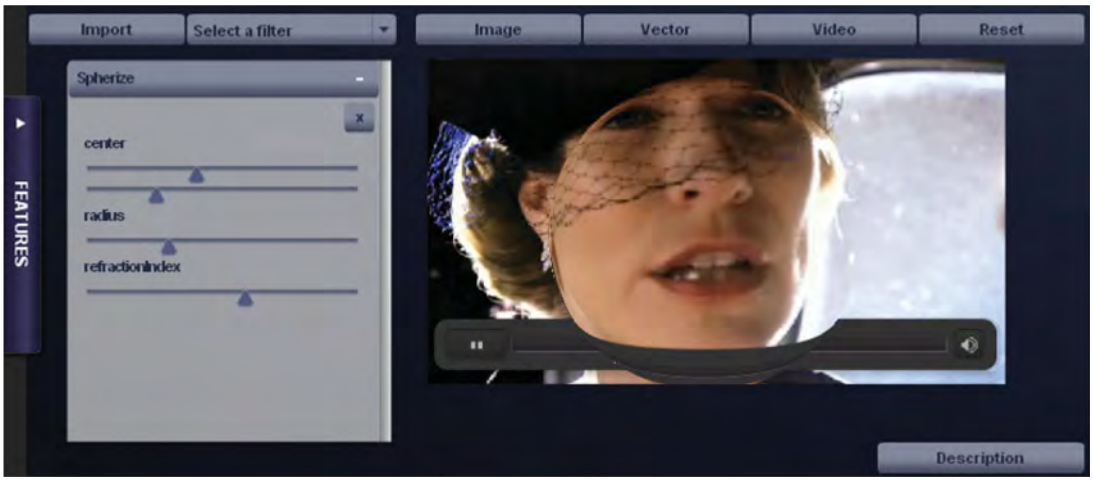

Figure 15: Custom filters and effects (e.g., Spherize distortion) are applied dynamically to a video in real-time using the Flash Player 10 features demo[23].

Video on the web is probably most familiar as an increasingly popular medium for news, sports, TV, concerts, and movies. Outside of entertainment, video is often used to support applications related to training, collaboration, surveillance, and social networking.

Both the quantity and quality of video available on the web has been steadily increasing over time, to the point now where it is beginning to encroach upon and rival traditional broadcast networks. Flash is contributing to this trend with its support of HD video, and increasingly, its support for GPU (graphics processing unit) hardware acceleration. The popularity of video across the web has been given a further boost with the Flash player's recently introduced support for hardware acceleration on a number of smartphones. Slowly, more and more applications are beginning to incorporate video, sometimes in new and previously impossible ways.

Immersive video[24] is one such example of utilizing video in innovative ways previously not possible. Users can freely pan in any direction throughout the entire video, a sensation similar to looking around by directing your gaze and rotating your head as if you were actually present within the scene being played back. There are many possible uses for this type of technology documented on the immersive website. The technology is particularly useful in route surveys and reconnaissance; for example imagine a periscope which records a full 360 field of view video in which many sets of eyes can independently scan simultaneously, whether in playback or live mode.

Figure 16: The "Boat Raid" demo by Immersive Media[24] combines video with interactive 360 degree panning.

RealityV[25] is another example of an application which uses Flash's input and multimedia capabilities to provide an immersive experience for users. The system combines immersive 360 video, head-tracking, and multi-channel surround audio with user interaction to provide the 'highest level of realism possible' in a situation simulator. Unlike avatar based simulators, the use of real actors conveys the subtleties of human gestures and expressions and can increase soldiers' awareness and knowledge of foreign cultures.

> *As a user navigates the environment, they are able to fully observe the live action content, which uses trained character actors and is recorded in high definition, digital media. Throughout the experience, trainees are able to provide input that impacts the scenario's outcome while maintaining vital metrics based on their performance. Favorable choices result in positive outcomes, while bad ones oftentimes take the story in an unfavorable direction with severe consequences.[25]*

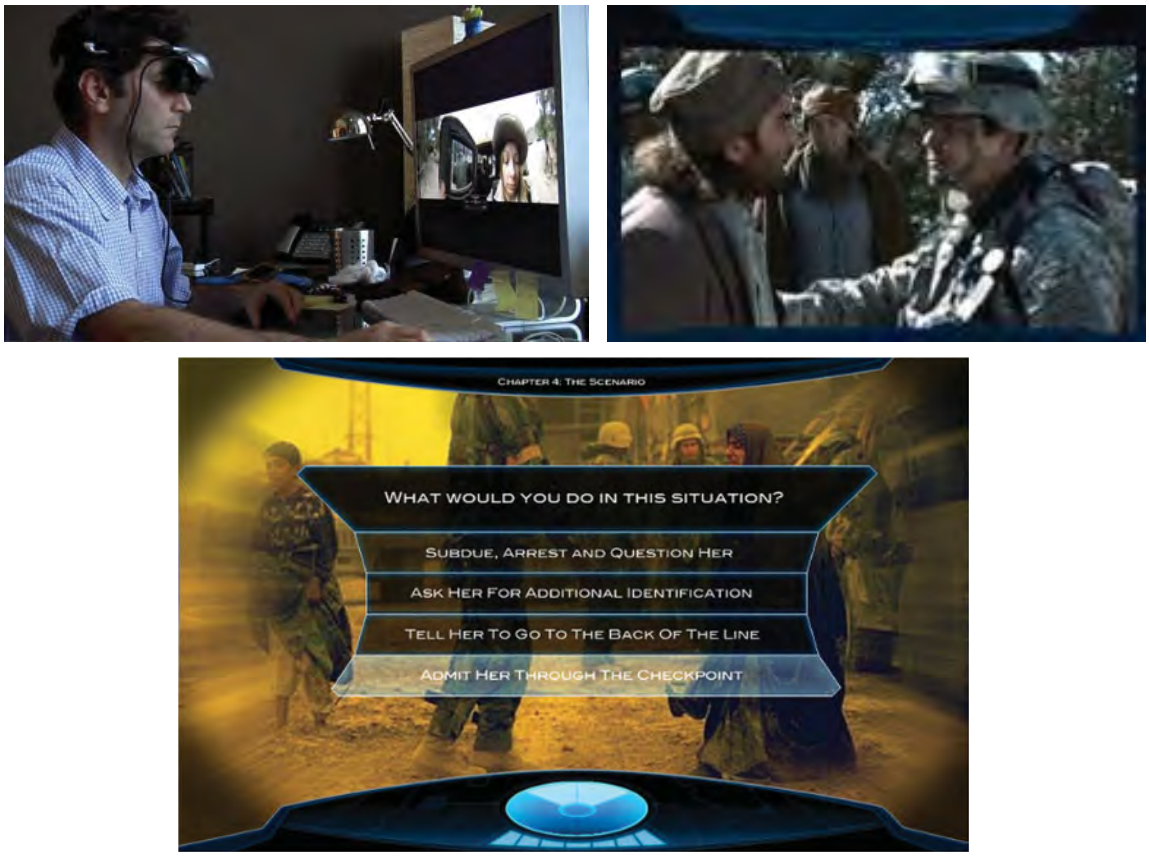

Figure 17: "Using live action presented through a custom developed Adobe Flash video delivery system, trainees can view and interact with media presented via 360-degree video with spatially correct binaural audio"[25].

Video related research at Adobe's Advanced Technology Labs[26] (ATL) has resulted in several noteworthy projects. One such research project is detailed in a recent ATL paper titled 'Video Object Annotation, Navigation, and Composition'[27] in which several novel means for interacting, analysing, and annotating video are explored. The ATL paper investigates unique video annotation capabilities which enable users to affix text or graphics to 'real' 2D objects within a video frame (Figure 18). The annotations stay attached to the target object and persist throughout the video duration (assuming the target object remains in the scene). The "Graffiti" feature adds planar perspective correction – annotations appear correctly oriented to any 2D surface. The research also deals with occlusion - "A rectangle created on the first frame sticks to the background even when its anchor region is partially or completely occluded. The annotation changes from yellow to red to indicate occlusion."[27]

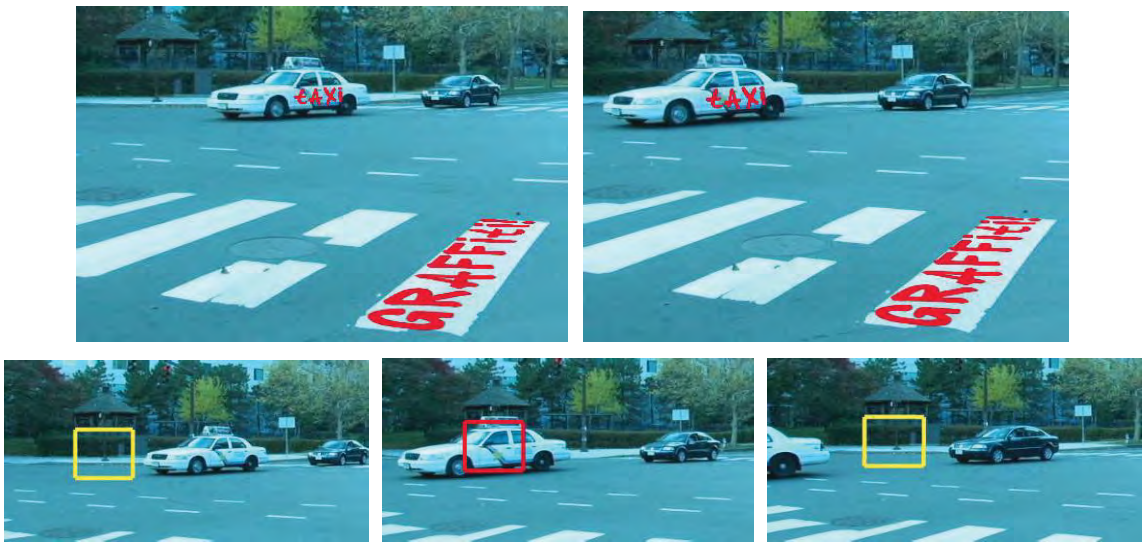

Figure 18: Text annotations affixed to video objects maintain planar perspective and remain attached (persist) throughout the video (Top). An annotation (rectangle) 'sticks' to its anchor region even when occluded (red rectangle) by other video objects (Bottom).

The ATL paper also introduces a video analysis tool known as 'path arrows' – a system to identify unique video objects and their motion paths. The motion track of an object within a video scene is then represented using a translucent yellow arrow (Figure 19). The generation of path arrows is made possible in part by combining computer based motion analysis and advanced object tracking.

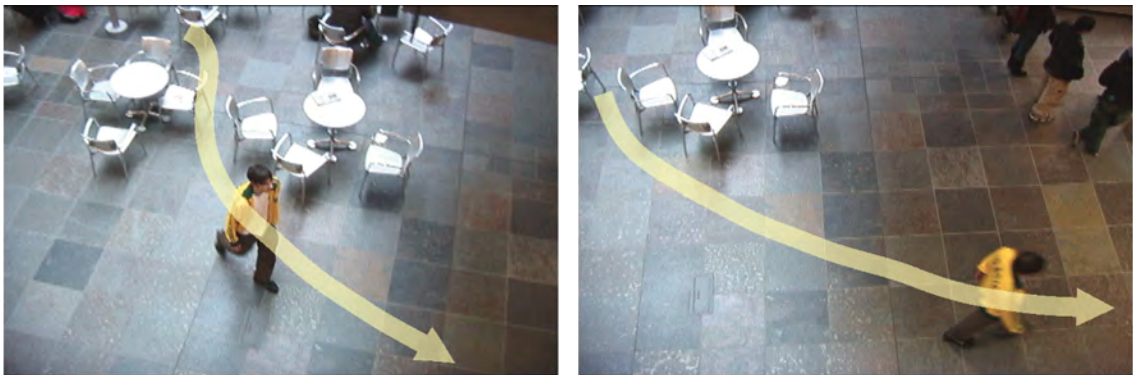

Figure 19: 'Path arrows' highlight the trail taken by a moving object captured on video.[27]

It is not hard to imagine the potential utility of such annotation and video analysis tools in many defence and security based applications (e.g., surveillance analysis, training, collaboration, and situational awareness). Again the Flash platform is well equipped for creating, deploying, and operating these proposed hybrid information systems which merge text, vector graphics, and video with unique interactivity.

Using ActionScript, it is also possible to enable unique interactions for controlling video playback. Using mouse gestures to control video playback[28] is one such interaction technique in which various input motions (e.g., dragging left) with a mouse or other input controller, trigger actions that would normally be done by clicking on traditional playback controls like buttons. The advantages with gesture based systems include more real estate for content and user control is not location specific on the screen (e.g., users do not have to hunt and position their pointer on a specific button).

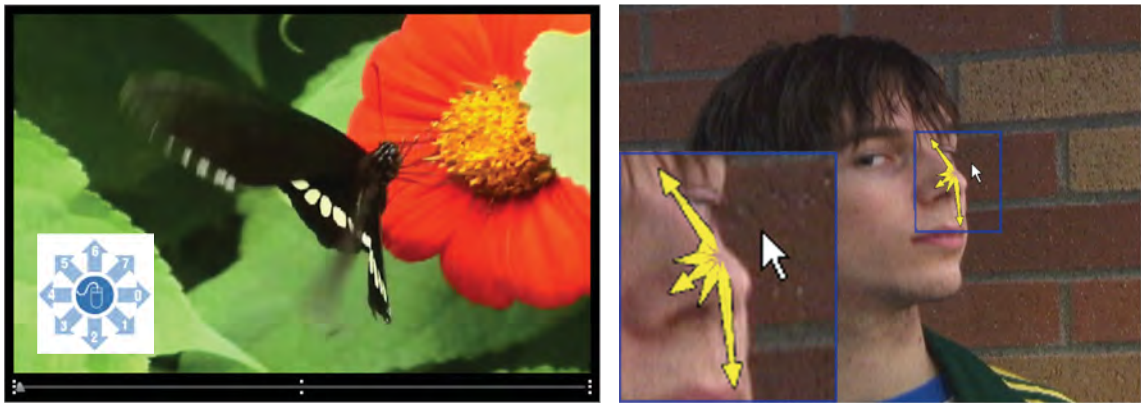

Figure 20: (Left) An interactive video uses mouse gestures to control playback[28]. (Right) Video navigation (scrubbing) using direct object manipulation[27].

Another novel technique for controlling video playback is video navigation (scrubbing) via direct object manipulation. The ATL paper investigates this direct manipulation method in which an object in the video can be clicked and dragged in order to traverse the video timeline. In this case the video object is a person's face and it is used with the mouse to navigate to video frames that

correspond with the facial positioning. A use case might be reviewing surveillance video with a suspicious person and finding frames with the best view of the suspect's face. Another example of direct object manipulation for scrubbing through playback is the Tactical Data Block demo[29] attached in Section A.1.10 of Annex A. In the map-centric tactical display, the track of a surface ship can be directly scrubbed to control the timeline of the entire tactical picture.

2.2.5 Animation

Animation typically refers to the motion of visual objects over time; however in Flash's case this definition can be extended to also include properties of the object itself. For example, a graphic object may change size, shape, opacity, color, or any number of attributes over time in addition to its positional properties.

Flash originated back in the mid 90s specifically as a vector based animation application called FutureSplash Animator. Since then Flash has steadily expanded in terms of core features, however animation remains a mainstay ability of the Flash platform. The animation capabilities of the player have blossomed over the years, in part driven by the popularity of the web, the increased capabilities of end user's hardware, and steadily improving connection bandwidths. At the same time the popularity of the player and its authoring tools have given rise to a very powerful, robust, and mature animation platform.

Flash animation is commonplace throughout the web, probably most familiar in casual online games, animated short films, and web advertising. Another more general animation example is the looping animations often used in meteorological forecasts for displaying track information, like the path of a hurricane over time. These are just a few examples of more common animation effects people are familiar with. However, what's more interesting in the context of this paper is how animation is leveraged in application development, especially within interface design and data visualization.

Animation is quite often employed in GUI design to subtly convey system feedback to the user. If you use any modern GUI based operating systems and applications you experience such animations repeatedly on a daily basis. If these animation effects were done correctly, chances are you would not specifically recall any of them. Over time the amount of visual effects employed in application interfaces has grown steeply, partly because of the increasing rendering power of client machines, and partly due to the ever increasing volumes and complexity of their underlying information systems.

Familiar examples include blinking alerts to indicate a change in application status, a spinning hour glass to convey busy, a growing progress bar for loading events, and the stealthy fade in and out animation of a new email notification.

Transitions from one state to another can be animated to emphasize the transition, such as the 'genie effect'[30] on the Mac OS for minimizing an application window. While emphasizing transitions between maximum and minimum window sizes, this animated effect also indicates the specific desktop location the application came from or went to. The 'CoverFlow' mode on the iPhone is another example that uses a carousel style animated menu for navigating music files.

Figure 21: The 'genie' animation effect[30].

Animation is also a powerful technique which can be used to enhance visualizations. Using Flash, the Gapminder organization offers an advanced visualization tool via their website for understanding large data sets related to world development and population trends.

> *"Gapminder is a non-profit venture promoting sustainable global development and achievement of the United Nations Millennium Development Goals by increased use and understanding of statistics and other information about social, economic and environmental development at local, national and global levels."*[31]

The visualization utilizes animation to reveal global population and demographic trends over time.

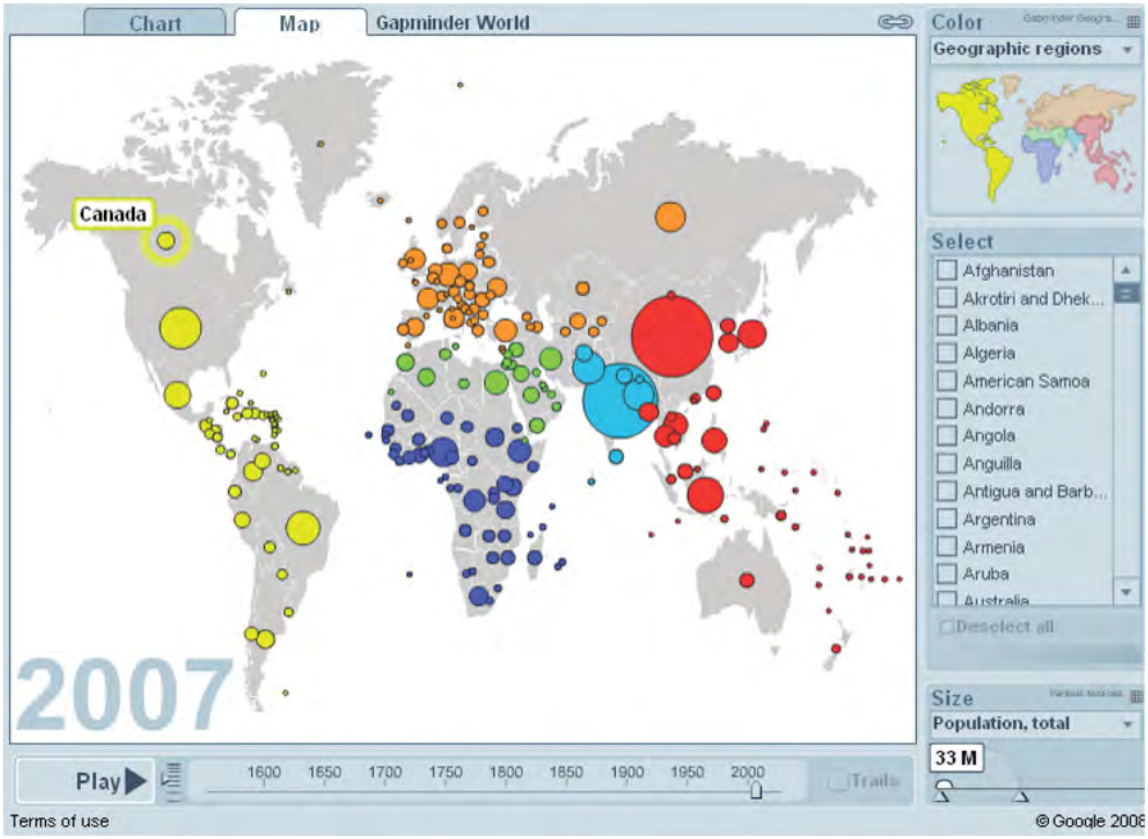

Figure 22: [Gapminder World][31] is an interactive Flash based visualization which uses animation to show world population and demographic changes over time.

The University of Utah's 'Cell Size and Scale'[32] visualization is an interactive Flash visualization which uses animation to provide a continuous context between visible sizes and those too small to see naturally. Beginning with items visible to the naked eye, users can zoom in to see smaller and smaller objects in a smooth and continuous manner.

Figure 23: The University of Utah's 'Cell Size and Scale' Flash based Visualization[32].

Animation in Flash can be accomplished in several different ways, depending on the tool used. Among the many choices in Flash authoring tools (Section 4), Adobe's Flash Professional IDE is easily among the most recognized and popular for projects involving lots of animation. This is not surprising given the history of Flash Professional as the original Flash authoring tool, and the fact Flash itself originated as a graphical animation platform. Most Flash developer oriented IDEs are

limited to runtime programmatic animation, in contrast to designer oriented tools which emphasize a design-time visual GUI for creating animations. Adobe's Flash Professional IDE is a hybrid tool when it comes to Flash animation by offering support for both basic animation workflows.

Using Flash Professional, design-time animation techniques are frame-based. Frame-by-frame animation is the 'old-fashioned' tedious method of incrementally moving (or modifying) an object for each and every successive frame. 'Tweening' on the other hand is a much more automatic animation method where the designer defines the start and end 'key' frames, and the software automatically generates all the intermittent frames in between – the 'tween' frames. Tweening can also be done programmatically at runtime using a tweening API such as Flash's built-in tweening API or any of several third-party APIs. Third-party tweening APIs typically offer improved performance, simpler usage, and more advanced features for tweening object attributes above and beyond what is available by default with Flash Professional. For example, GreenSock's[33] tweening APIs offer significantly improved performance and many advanced tweening features not available in the native Flash tweening API. Section A.1.3 of Annex A includes an interactive tweening demo application based on the GreenSock API.

'Easing' is a tweening specific term which essentially refers to the speed and acceleration characteristics of a tween animation. For example, a 'bounce' easing function applied to a ball's (tween) animation would give it bounce behaviour. Like tweening, developers can use Flash's built-in classes for easing or chose from several third-party alternatives.

Flash Professional CS4 introduced a new animation tool for implementing inverse kinematics (aka 'the bone tool'). Using this tool, armatures (aka 'bones') can be added to 2D animation objects and connected (via joints) into a skeletal frame. The object's underlying skeleton can then be easily 'posed' using direct manipulation in Flash Professional CS4's GUI. Lee Brimelow's inverse kinematics animation web tutorial[34] provides a great overview and introduction to Flash animation using the bone tool .

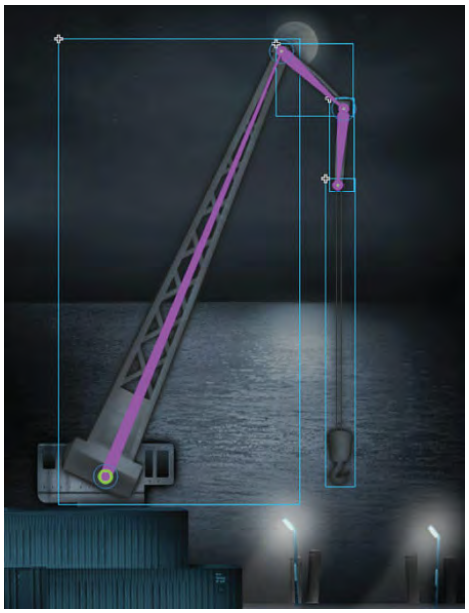

Figure 24: The inverse kinematics (aka'bone') tool in Flash Professional CS4[34] .

2.2.6 Text

Computer interfaces have slowly been evolving away from their purely text based origins to more graphical based paradigms. The reasons for this are varied but are at least partially attributable to ever advancing progress in display, input, and performance capabilities. Yet despite the trend toward richer GUI designs sporting more and more graphical controls, visualizations, and multimedia, text remains an integral and central part of the way we interact with applications. This is even truer for web based applications operating under low bandwidth conditions.

Flash has a text engine which can be leveraged (and extended) using Adobe's open source Text Layout Framework (TLF). Adobe summarizes some of the key features of the Flash player's text engine as[35]:

- *Bidirectional text, vertical text and over 30 writing systems including Arabic, Hebrew, Chinese, Japanese, Korean, Thai, Lao, the major writing systems of India, and others*
- *Selection, editing and flowing text across multiple columns and linked containers, and around inline images*
- *Vertical text, Tate-Chu-Yoko (horizontal within vertical text) and justifier for East Asian typography*
- *Rich typographical controls, including kerning, ligatures, typographic case, digit case, digit width and discretionary hyphens*
- *Cut, copy, paste, undo and standard keyboard and mouse gestures for editing*
- *Rich developer APIs to manipulate text content, layout, markup and create custom text components.*

The complete list of features available using the TLF framework is extensive and included with the TLF release notes[36]. The '[World Class Text Tour]'[35] demo application available on the Adobe labs web site showcases several of these advanced text capabilities.

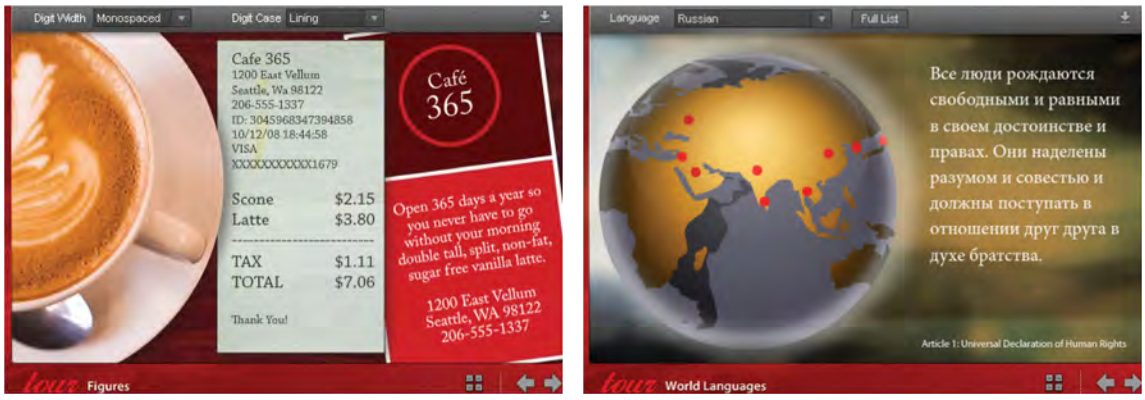

Figure 25: The '[World Class Text Tour]'[35] web app demonstrates some of the advanced features of the Flash text engine.

Text in Flash can be stretched, flipped, rotated, twisted and basically distorted in every imaginable way. Applying special effects to text such as transparency, 3D, glow, shadows, and other filters is also possible. The Adobe 'Drawing Text' application demonstrates animated text following a user defined drawing path[37].

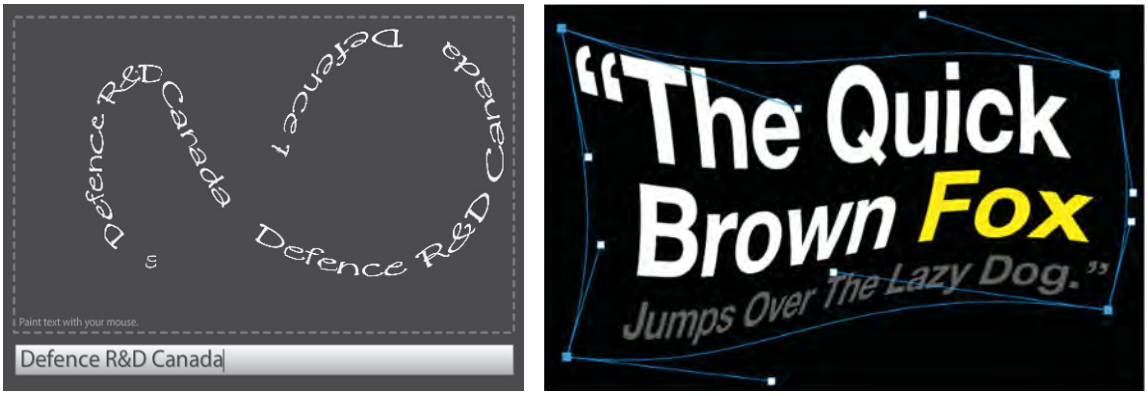

Figure 26: Examples of runtime text distortions using the Flash player's text engine. (Left) Animated text following a user drawn path[37]. (Right) Interactive runtime distortion of text.

A good example of software which makes heavy use of the Flash player's text rendering capabilities is the NY Times 'Times Reader 2.0'[38] application (Figure 27) .

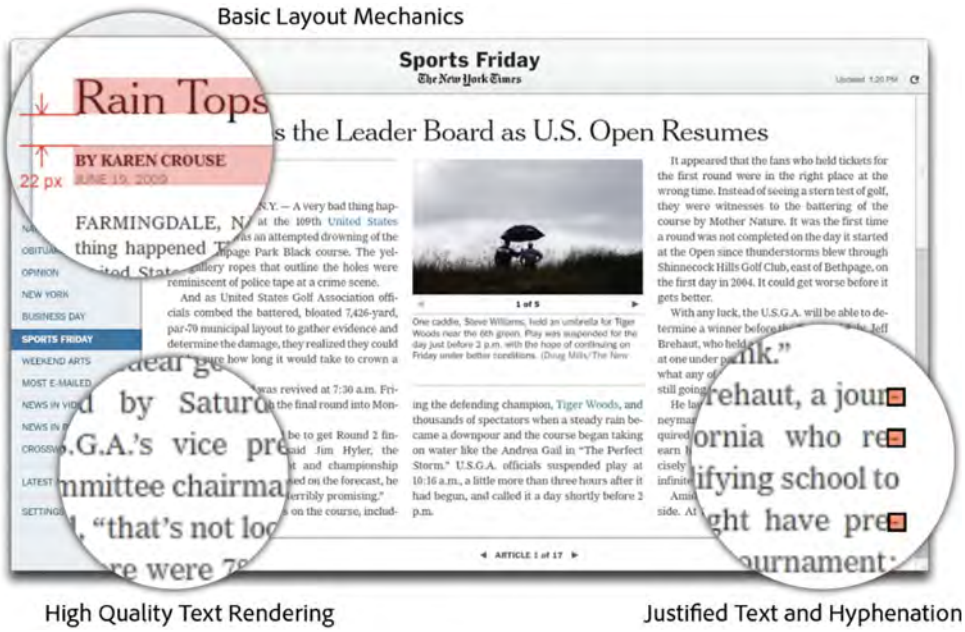

Figure 27: The NY Times 'Times Reader 2.0'[38]application takes advantage of the advanced Text rendering features of the Flash player.

Beyond appearance, Flash's animation capabilities can also be applied to text including scrolling, flipping, rotating, fading, and blinking to name a few possibilities. Dynamic text effects are commonly used for alerts, visualizations, and to conserve screen real estate (e.g., scrolling news ticker). For example, the Track Data Block demo included in Section A.1.10 of Annex A features scrolling text in one of its tool-tip designs.

Finally, and depending on the application and context, text 'behaviours' can be programmed using ActionScript. For example, Adobe recently released the "Squiggly" spell checking API for Flash based applications. Many small or embedded devices which lack a full-size keyboard utilize predictive text engines to improve the ease and speed of text input. Such tools dynamically propose completed words and phrases based on the user's initial input character(s). The Flash Lite player supports all the major predictive text engines (such as T9, eZiTap/eZiText, and iTap)[39] and third-party tools[40] are available for the DIY crowd. Future text behaviours could also arise with new inputs like multi-touch; one example might be a touch surface virtual keyboard which responds to 'hard' presses by capitalizing the letter pressed.

2.2.7 Interactivity

A longstanding strength of the Flash platform is its extensive support for interactivity. Essentially interactivity refers to the ability of an application to respond to user input in a pre-programmed manner (behaviour). In other words how different GUI display objects respond to various user triggered events (e.g., mouse click, keyboard press). Developer-oriented tools tend to offer numerous pre-made GUI controls (e.g., buttons and textboxes) with support for customization. However, Flash design-oriented tools such as Adobe's Flash Professional CS4 offer full-featured graphical design tools for creating custom UI controls from scratch. Importantly, all of the graphics, animation, advanced text, and multimedia capabilities discussed previously can be easily enhanced with interactivity.

Regardless of graphical design, the key enabling technology for all user interaction with display objects in Flash is its native programming language ActionScript. ActionScript in its current incarnation (version 3) is a powerful and extensible object-oriented programming language. Using ActionScript, developers can create, share, and apply interactive behaviours as desired. In the same manner as GUI design, programming interactivity can range from simple standard data operations to highly complex interface behaviours created completely from scratch. More information regarding ActionScript as well as other Flash development languages can be found in Section 2.2.8 'Application Development' and Section 4 'Developer Tools'.

Examples of some of the unique interactions that can be created using Flash can be experienced using the demos available on the PowerCursor[41] website. PowerCursor is a Flash software development kit (SDK) for creating 'optically simulated haptic feedback', or in other words, simulating sensations like stickiness, gravity, slipperiness, and roughness with the mouse pointer. A good example from the web demos is the 'slick-disabled' menu system which 'repels' the pointer away from disabled menu items (Figure 28). Like a lot of Flash examples it is difficult to describe and much easier to understand by simply trying. The example uses a simple grocery item list, but the menu items could just as easily be anything else (e.g., vessel classes).

Figure 28: The 'slick-disabled' menu created using the PowerCursor SDK[41] aids the navigation and selection of menu items.

Another simple example of unique interactivity using Flash is the Mouse Gesture Demo[42] by Didier Brun. The experimental web application recognizes specific mouse movement sequences (gestures) and translates them into text characters. Gesture based interactivity can be useful for experimenting with alternate system interfaces such as those without a traditional keyboard (e.g., touch driven interfaces).

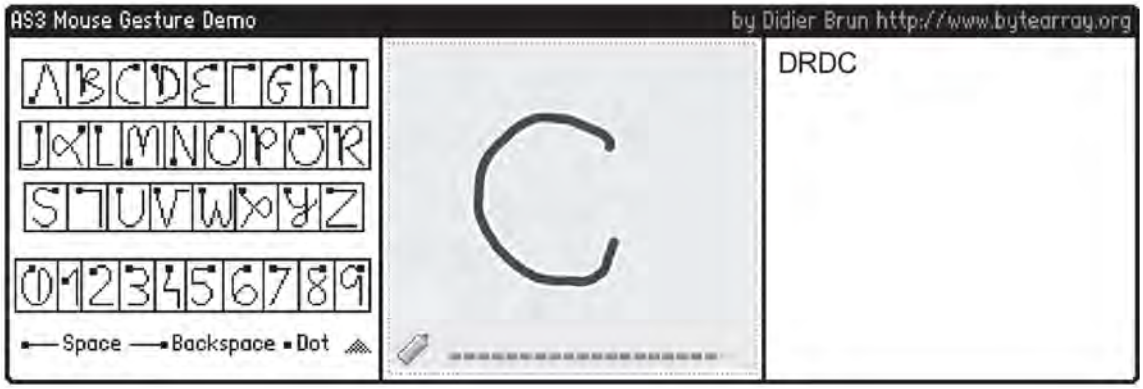

Figure 29: The Mouse Gesture Demo[42] recognizes specific mouse movement sequences and translates them into text characters.

Most (if not all) of the applications described throughout this document also testify to the interactive capabilities of Flash, however Section 2.2.8 'Application Development' and Section 3.4 'Human Computer Interface Research' offer particularly relevant examples. Furthermore readers can experience interactivity firsthand using any of the demos included at Annex A as they are all interactive.

Finally, it is important to consider input hardware support in any discussion of Flash's interactive capabilities. Mice and keyboards are the standard input hardware on most desktop systems, and Flash support for these devices and their specific input capabilities is well established and mature. One odd and notable exception is the lack of custom behaviours when right-clicking a mouse – invoking a context menu is the only supported behavior. Other input devices with native Flash player support include web cameras, microphones, and more recently multi-touch devices (see Section 3.10). In addition, Adobe has also been steadily adding Flash support for smartphone specific capabilities such as GPS, compass, and accelerometer. Beyond these more popular and familiar input peripherals, there is also a growing myriad of digital devices and peripherals with new and unique input capabilities. Many of these devices are also accessible to Flash developers using either native or third-party APIs. Section 3.4.2 describes a few recent and noteworthy examples of novel input hardware supported by Flash.

2.2.8 Application Development

The Flash player is much more than a multimedia and graphic animation renderer; it is a very capable application runtime environment for hosting the execution of Flash applications. These programs can run within the web browser using the Flash browser plug-in or as desktop applications via Adobe's integrated runtime (AIR). Flash applications are typically constructed based on the familiar (web) client-server architecture. The Flash player itself is a client-side technology, and the Flash platform has historically been heavily focussed towards the client. That said, the platform has definitely evolved on the back end, particularly with regard to server-side video and data services solutions. This also includes a growing list of data connection, transfer, streaming, binding, and remoting technologies for sharing data and content between servers and clients.

Flash applications are typically programmed using ActionScript, the native programming language of Flash. ActionScript is a powerful object-oriented programming (OOP) language which compiles into platform agnostic byte code for the Flash player's program execution engine – the ActionScript virtual machine (AVM). In other words, the compiled Flash applications are completely cross-platform ready and will run (via the AVM) on any device or OS that has the Flash player installed including Windows, Linux, Mac, and countless mobile devices. If this cross-platform application development paradigm (e.g., write once, run anywhere) sounds familiar, it should – it is the essence behind the java platform. If you imagine combining the java runtime engine (JRE) with a multi-media player (e.g., Apple QuickTime), throw in a graphic animation renderer, and you will have a general picture of the Flash player. It is the 'Swiss army knife' equivalent of web browser plug-ins.

Programming in Flash can be done in a scripting, procedural, or object-oriented (OOP) style, and it is not unusual to see combinations of all these techniques within a single application.

ActionScript provides all the standard programming elements you would expect in any modern development language including primitive and complex data types, looping constructs, conditional expressions, functions, arrays, and operators. ActionScript 3 (AS3) OOP-specific capabilities include classes, packages, inheritance, encapsulation, interfaces, and polymorphism. What is more unique to Flash are the countless native APIs specific to AS3 and the Flash player including specific class libraries for graphics, animation, images, video, UI controls, audio, and interactivity to name a few. Each of these built-in classes will in turn have their own unique set of properties and methods in addition to the properties they inherit from their parent classes. For example the Graphics class has methods for drawing lines, rectangles, and squares while the Sound class has load, play, and close methods.

2.2.8.1 Flex

For Flash developers, Adobe's open source 'Flex' framework is the tool of choice for building data driven (e.g., business) applications, especially those with sophisticated user interfaces and rich interactivity. "The Flex framework provides the declarative language, application services, components, and data connectivity developers need to rapidly build rich Internet applications (RIAs) for the browser or desktop"[43]. Flex is basically Flash for application developers, and the term Flex is often used as a prefix to categorize and distinguish tools used specifically for Flash programming and application development.

The declarative language included with Flex is mxml (no official acronym meaning) – an xml language used to lay out user-interface components in Flex applications. Mxml is to some degree an alternative language to ActionScript, appealing to web developers more familiar with using tag based languages. In terms of application programming capabilities, mxml is a limited subset of ActionScript; coding of business logic and program flow control must still be done using ActionScript. The relationship between mxml and ActionScript can be roughly equated to the link between html and JavaScript; the markup languages are great for adding and laying out interface components, but a programming language remains necessary for adding interactivity and functionality. The upcoming version 4 of Flex adds a new drawing specific dialect of mxml called fxg (Flash xml graphics). Again as with mxml in general, fxg is a tag-based (declarative markup) alternative to ActionScript that developers can use to draw Flash graphic objects and skin components.

The Flex framework contains a myriad of pre-built UI components; everything from buttons and check boxes to advanced data grids and rich text editors. Included in the Flex library are reusable classes for local file system and database operations, network communication, interface layout and behaviour, and many more common programming tasks.

Adobe's 'Tour de Flex'[6] demo is without doubt the quickest and easiest way to gauge application development capabilities on the Flash platform. The browser version of Tour de Flex can be experienced online, or the desktop version can be downloaded and installed for experiencing additional desktop specific features. The demo contains numerous working development samples using a variety of Flex components. The samples are organized in a logical tree hierarchy (Figure 30) and include categories for UI controls, cloud APIs, data visualization, mapping, data access, and many more. In addition to viewing and interacting with each sample, developers can see the source code used.

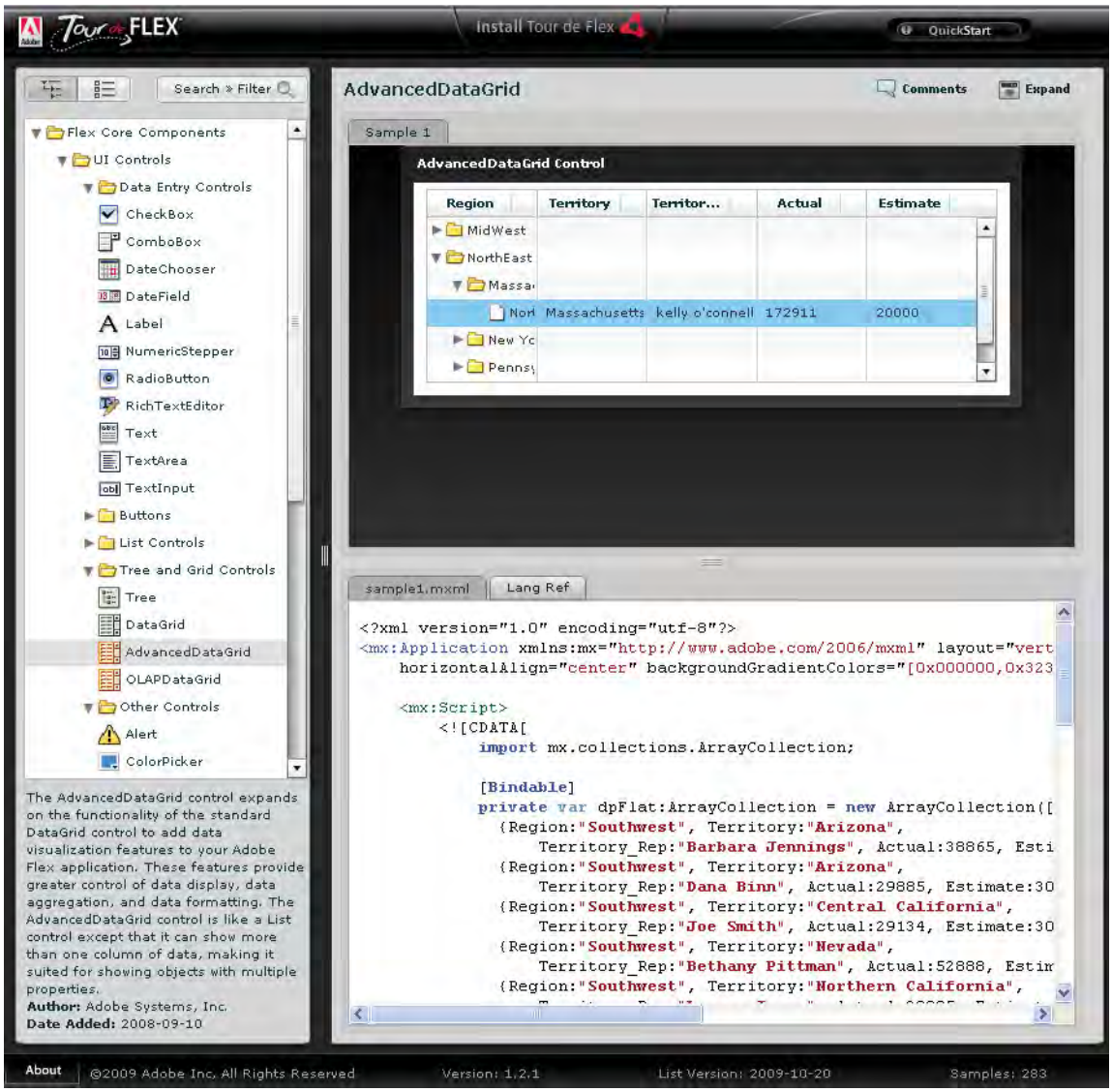

Figure 30: The '[Tour de Flex](#)'[6] demo provides an easy way to sample application development capabilities on the Flash platform.

The application development features available to Flash developers will largely depend on the tool(s) being utilized. The primary Adobe based tools for building applications are Flash Professional (currently version CS4) and Flex Builder (currently version 3). Generally the development process will begin with the addition and layout of pre-built UI components, followed by programming their interactive behaviours based on business logic rules, as well as binding them to data as applicable. Flash Professional includes a small assortment of familiar user interface controls out of the box (Figure 31). Included are everything from simple buttons, text input boxes, and labels to color pickers and data grid controls. However, the Flex Builder IDE includes the Flex framework which in turn includes a much more extensive set of UI controls including tree, advanced data grid, and date picker controls. This is not surprising since Flex Builder is a developer oriented tool and developer applications tend to be more functional, data-

driven, and task oriented (e.g., updating records in a database). Refer to Section 4 for a more complete discussion of Adobe and other third party software tools for Flash.

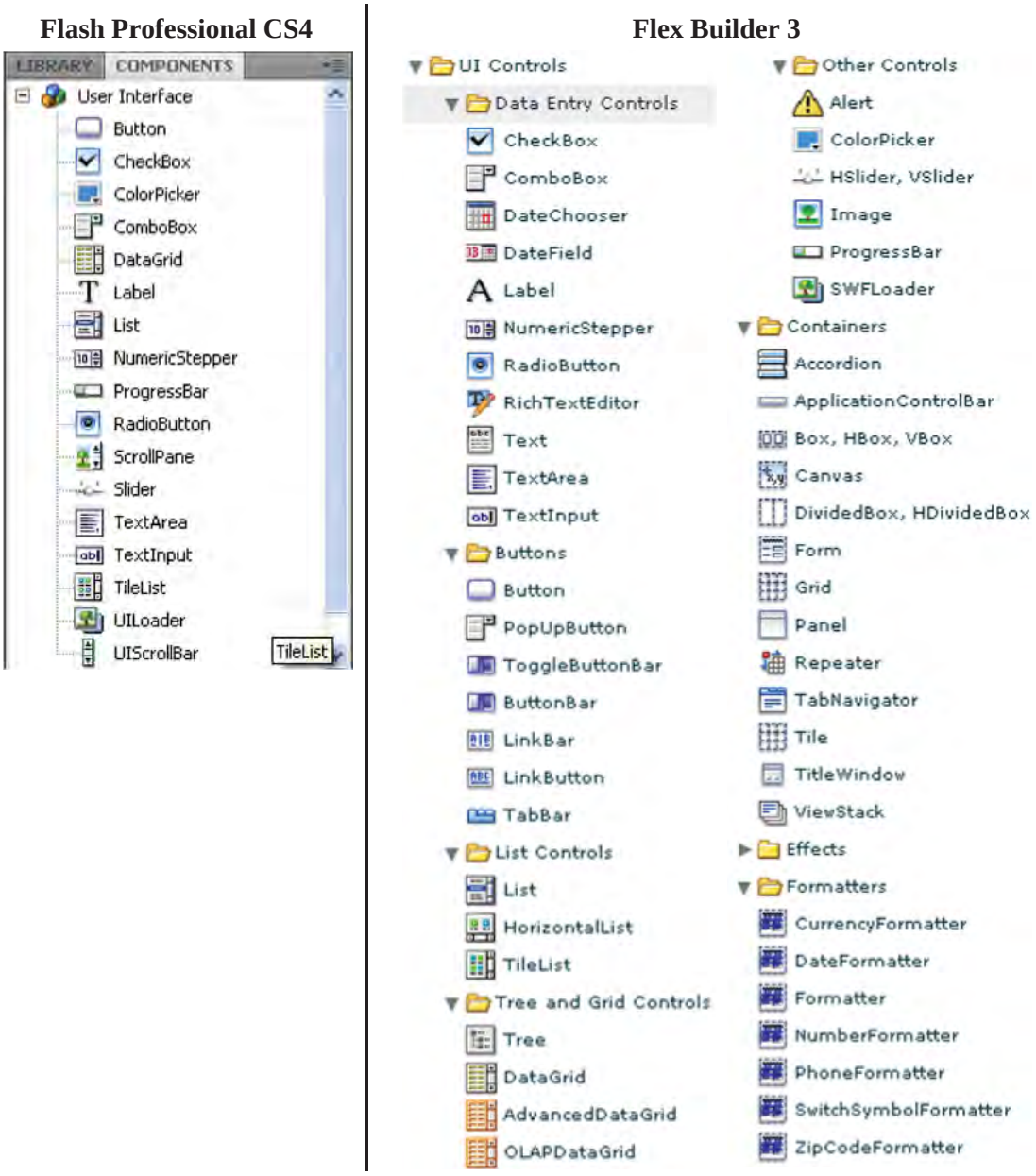

Figure 31: The selection of out-of-box UI controls available from Flash Professional CS4 (left) and Flex Builder 3 (right). The extensive list of Flex UI controls reflects the fact it is more of a developer oriented tool geared towards building data-driven applications.

Depending on the Flash development tool, you can customize (e.g., skin) components or build them entirely from scratch. Design tools such as Flash Professional are particularly well-suited for creating entirely new components which do not resemble anything that already exists. For example, Flash Professional's GUI based drawing and animation tools can be used to quickly

create new kinds of UI controls. However any significant programming (e.g., for interactivity) might be better done using fully-featured, developer focused IDEs since the code editor in Flash Professional lacks many of the enhanced coding features common in most modern coding tools. Thus to reiterate, Flash creative design tools combined with the ActionScript programming language enable the development of new UI components with unique form and function.

2.2.8.2 Rich Internet Applications (RIAs)

It is easy to forget the humble web browser was initially designed as a simple and lightweight application for retrieving and displaying static html pages from a web server. Traditional browser interaction uses a basic page-centric request/response model; the user requests a page of content and the (entire) page is retrieved and displayed (Figure 32). The technology behind the web browser was quite simply never intended, nor designed, as an application runtime environment. Web applications which strictly comply with the html standard thus offer very basic interactivity and lack immediate feedback and responsiveness for users. Early web applications which featured a series of web forms, each with a single submit button are a classic example; all the business logic and user input validation was done on the server side one page at a time AFTER the page's submit button was pressed. This lack of immediate or real-time feedback meant no prevention of input error and no feedback to the user until after a complete input page was submitted, processed, and the results returned to the browser. The entire workflow is slow and frustrating for end users, not to mention the extra bandwidth due to both erroneous input and overly frequent information exchanges between the client and the server.

Regardless of the obstacles, there has nonetheless been an unrelenting and steady trend toward web applications, and thereby forcing the browser to act more and more like an application runtime environment. For developers there is the simplicity of developing and deploying applications for a single browser client which inherently supports cross-platform. For end users, web applications provide ubiquitous accessibility – accessing and using an application is no longer constrained to a particular device, place, or time. It is these fundamental advantages of browser based applications that have simply outweighed any backward compromises in the overall user experience.

That said, users have nonetheless become accustomed to the sophisticated GUIs, responsiveness, and rich interactivity commonly experienced with installed desktop applications. Consequently, web browsers have been struggling to transition from hypertext viewer to application platform. Resolving these issues and bringing web applications up to the same level of responsiveness and usability as installed desktop applications is the goal of the 'Rich Internet Applications' (RIA).

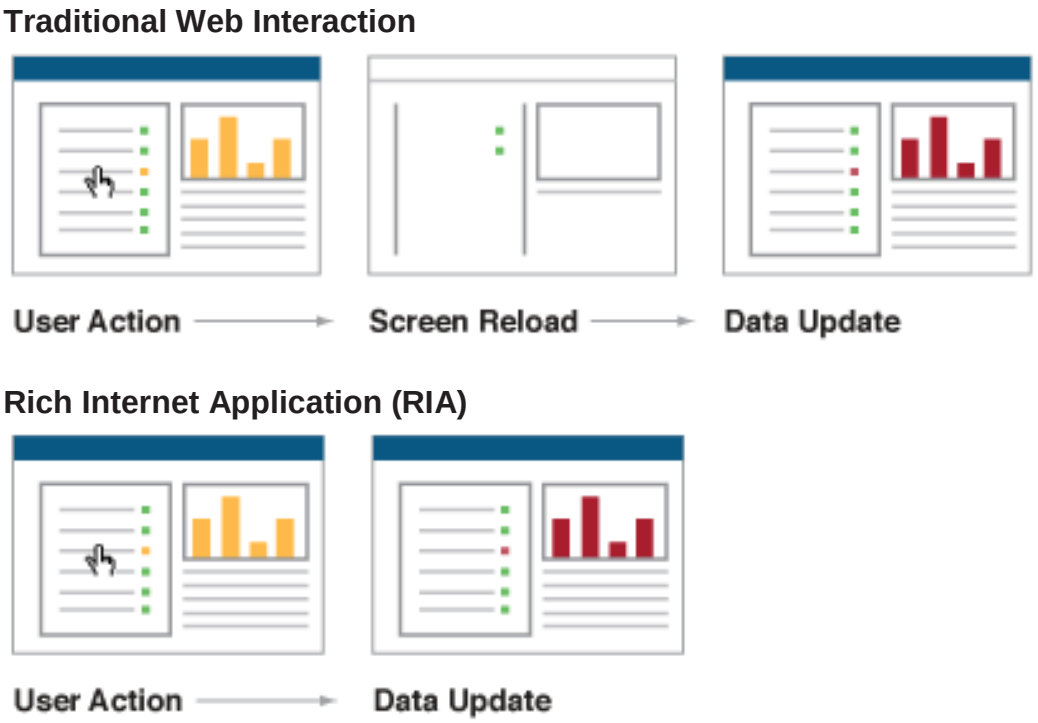

Figure 32: Traditional web interaction (top) is page centric and slow compared with the data driven interaction of RIAs(bottom)[44].

RIA is the term commonly used to describe the evolving hybrid class of software which combines the responsiveness and rich interactivity features of desktop programs with the accessibility and connectivity of web applications. RIAs are "web applications that have the features and functionality of traditional desktop applications"[45]. Additionally, RIAs offer improvements in scalability, reliability, and security. They generally take the form of sophisticated, highly interactive, data driven applications delivered over the web (Figure 32). In terms of architecture, RIAs reside between thin (pure) html clients (where all processing tasks are performed on the server) and thick local desktop programs in which all processing is performed on the local host (Figure 33). "RIAs generally split the processing across the Internet/network divide by locating the user interface and related activity and capability on the client side, and the data manipulation and operation on the application server side"[46].

To boost the richness and usability of *browser based applications* to the same level as their installed desktop equivalents, two basic RIA architectural strategies have emerged: browser *built-in* and browser *plug-in*. The built-in browser approach uses 'native' browser technologies like "Asynchronous JavaScript and XML" (AJAX), cascading style sheets (CSS), and extensible hypertext markup language (xhtml). By definition the browser built-in strategy to building RIAs avoids any dependency on browser plug-ins. Conversely, the plug-in based RIA architecture requires users to download and install a plug-in for their web browser.

Within the browser plug-in based approach to RIAs, Adobe Flash and Microsoft Silverlight are the predominant competing options. Of these two, Flash is the more ubiquitous, present on over

98% of internet users' computers regardless of OS[3]. Silverlight penetration is about 50% on Windows and Mac based computers, and currently not available for other operating systems. Flash is also significantly more mature dating back to the mid 90s while Silverlight did not appear until 2007. Silverlight does enjoy a huge developer community and some of the most popular and favoured design and development tools.

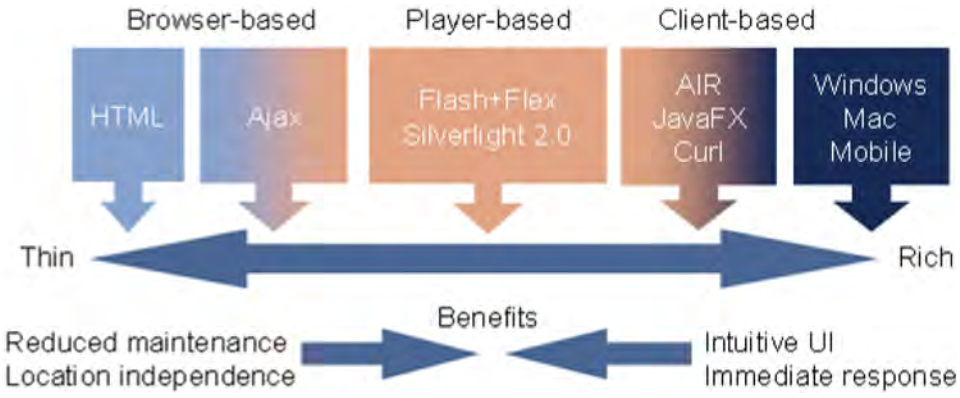

Figure 33: The spectrum of internet application architectures spans from simple HTML web pages, through increasingly capable RIAs, to internet enabled and locally installed desktop programs.

The final approach to RIA architecture forgoes the browser entirely in favour of a desktop installed runtime environment. This is precisely the model used by the java runtime engine (JRE), Adobe's integrated runtime (AIR), and Curl's runtime environment (RTE). This is by no means a 'lightweight' solution, as users have to download and install a runtime environment which is typically an order of magnitude larger than a browser plug-in. Unlike web applications which are seamlessly loaded and run by the browser on the fly, each desktop RIA must be downloaded and installed prior to executing. Despite the inconveniences, installed RIAs offer distinct advantages simply not possible with web applications. For example, installed RIAs can operate offline, ideal for occasionally connected devices such as laptops and smartphones. They can also access local machine resources like the file system (e.g., saving user data locally). Other advantages include the ability to launch other installed applications, and generate alert notifications using the system tray.

Regardless of the underlying architecture, web applications that break-out of the aging page-centric interaction model are popping up across the web with ever increasing frequency and variety. Several RIA examples are provided throughout this paper, while countless more can be found online. A very relevant example in the context of this paper is the Virginia Interoperability Picture for Emergency Response (VIPER)[47]. VIPER is a map-centric situational awareness web application for emergency responders and decision makers. Using VIPER, emergency officials across multiple disciplines and jurisdictions can establish and maintain a common operating picture using a myriad near and real-time data layers for weather, traffic, police, fire, and many other information feeds.

> *In addition to providing a Web-based common operating picture and analysis tools, VIPER integrates with numerous information systems and links with approximately 250*

data feeds. Emergency commanders; first responders; and police, fire, and government officials can tap into a single information resource for better decision making.[48]

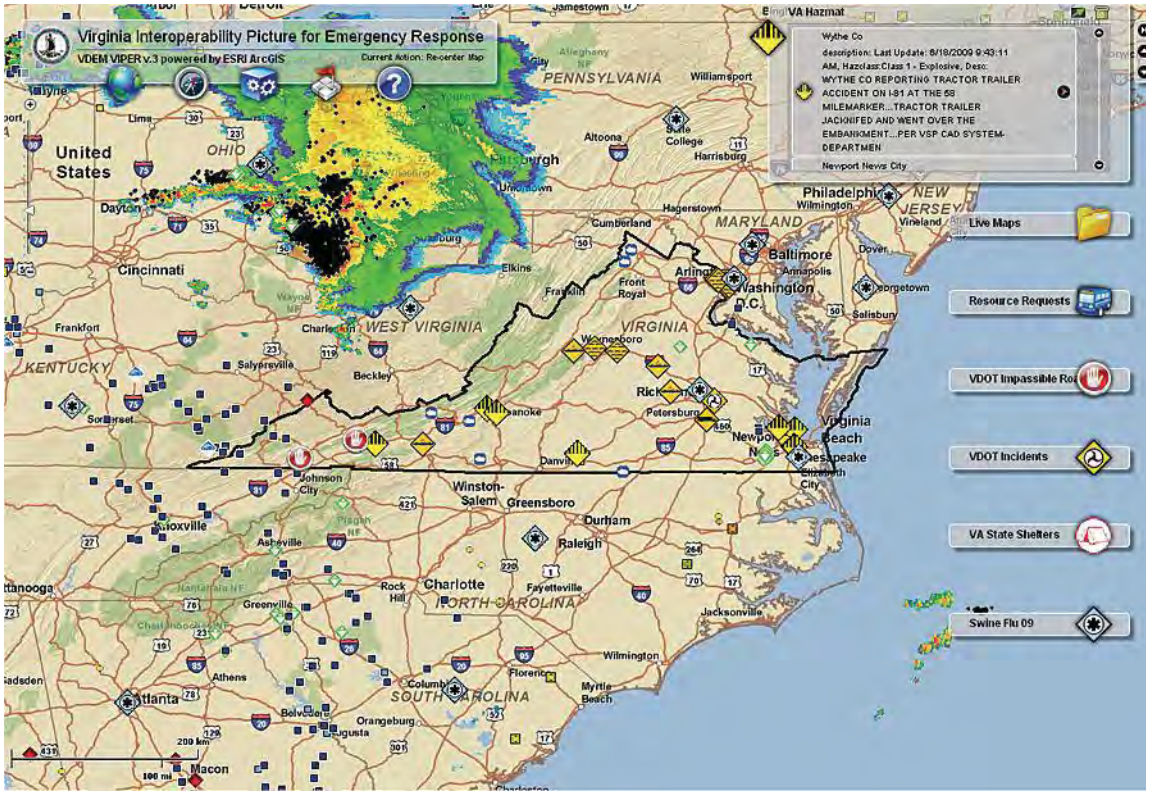

Figure 34: The Virginia Interoperability Picture for Emergency Response (VIPER)[47] is a map based, situational awareness Flex application with near and real-time layers for weather, traffic, police, fire, and many other situational awareness tools.(Image from [48])

2.3 The Flash Player

In laymen's terms, the Flash player is a relatively small browser plug-in (about 1.8Mb on Windows systems) that people use for playing multimedia (e.g., animations, video, and audio) as well as running web applications. From a software architecture perspective, it is a client-side runtime environment for executing Flash .swf (pronounced 'swiff') files. In fact, .swf files are the only type of files the Flash player can execute directly.

The history of the Flash player can be traced back to a vector graphics animation program called FutureSplash Animator, created by a small start-up company known as FutureWave back in the mid 90s (just as a new phenomenon called the internet was starting to bloom). Shortly thereafter, in December 1996, Macromedia acquired FutureWave and FutureSplash Animator was renamed

Macromedia Flash 1.0. In December 2005 Macromedia was purchased by Adobe including the Flash player and its growing suite of related development tools and technologies.

Over the years the player has steadily progressed through several successive versions up to and including the current version 10. With each new version the Flash player has adopted new capabilities, particularly in the areas of vector graphics, multimedia, animation, and programmability. The exact feature set is extensive and continually being updated (Figure 35). Section 2.2 reviews some of the core capabilities of the Flash player along with relevant examples.

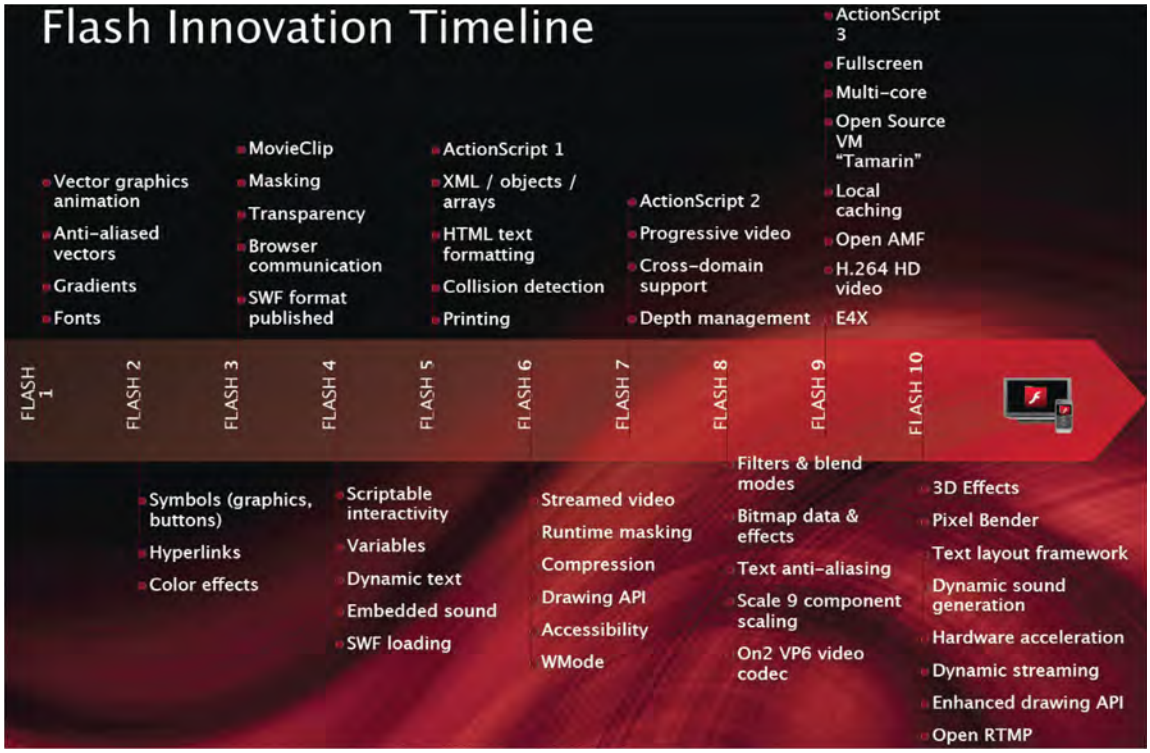

Figure 35: The Flash player has steadily evolved over time with increasing capabilities relating to graphics, animation, multimedia, and application development (Image from [49]).

The Flash programming language 'ActionScript', along with the corresponding ActionScript virtual machine (AVM) was introduced in version 5. ActionScript matured from a scripting language in version 1, through a procedural language in version 2, to a comprehensive object-oriented programming (OOP) language in its current version 3. AS3 was actually more of a complete overhaul than an evolution, as it required the addition of a second and entirely new ActionScript virtual machine (AVM2) to the Flash player. AS3 has provided the foundation of advanced application development in Flash and is a key enabler in Flash's emergence as a frontrunner platform for sophisticated second generation web applications.

2.4 Flash Lite

A lightweight version of the Flash player known as 'Flash Lite' was introduced in 2004 specifically for lower end devices which lacked the hardware resources to run the full Flash player. Not surprisingly the capabilities of Flash Lite are, as a rule, significantly less than its full version parent. The current version of Flash Lite is version 3.1 and feature wise is comparable to version 8 of the full Flash player, but also offers many video playback features found in version 9. However, this gap is diminishing, partly due to the increasing compute power of smartphones, as well as Adobe's effort to re-unify the Flash player across all platforms as of version 10.1.

Figure 36: An estimated 1 billion devices are equipped with the Flash Lite player[50].

Mobile phones have been the most popular Flash Lite target device by far, with an estimated 1 billion such devices equipped with the Flash Lite player as of 2009[50]. Flash Lite runs on a variety of mobile platforms including Windows Mobile, Java ME, BREW, and Symbian.

2.5 Adobe Integrated Runtime (AIR)

In the last couple years Adobe has been actively extending Flash applications beyond the browser onto users' desktops via the Adobe Integrated Runtime (AIR) technology. AIR is a cross-platform **desktop** runtime environment which enables web applications to be developed, deployed, and run as desktop installed applications. In additional to the Flash player itself, AIR includes a database (SQLlite), a web browser engine (Webkit), and several other desktop specific tools. Adobe market AIR primarily as a technology for desktop RIAs, in other words AIR is intended mainly as a platform for data-driven web applications **installed** on users' local machines. For developers, AIR is a 'write once, run anywhere' (WORA) solution in the same manner as the Java Runtime Environment (JRE).

> *"We solved the cross printer problem with Postscript; we solved the cross word-processing program problem with PDF. And with Flash, we solved the [problems of] multimedia and video incompatibility across the web. The next frontier is the cross operating system application runtime."[51]*

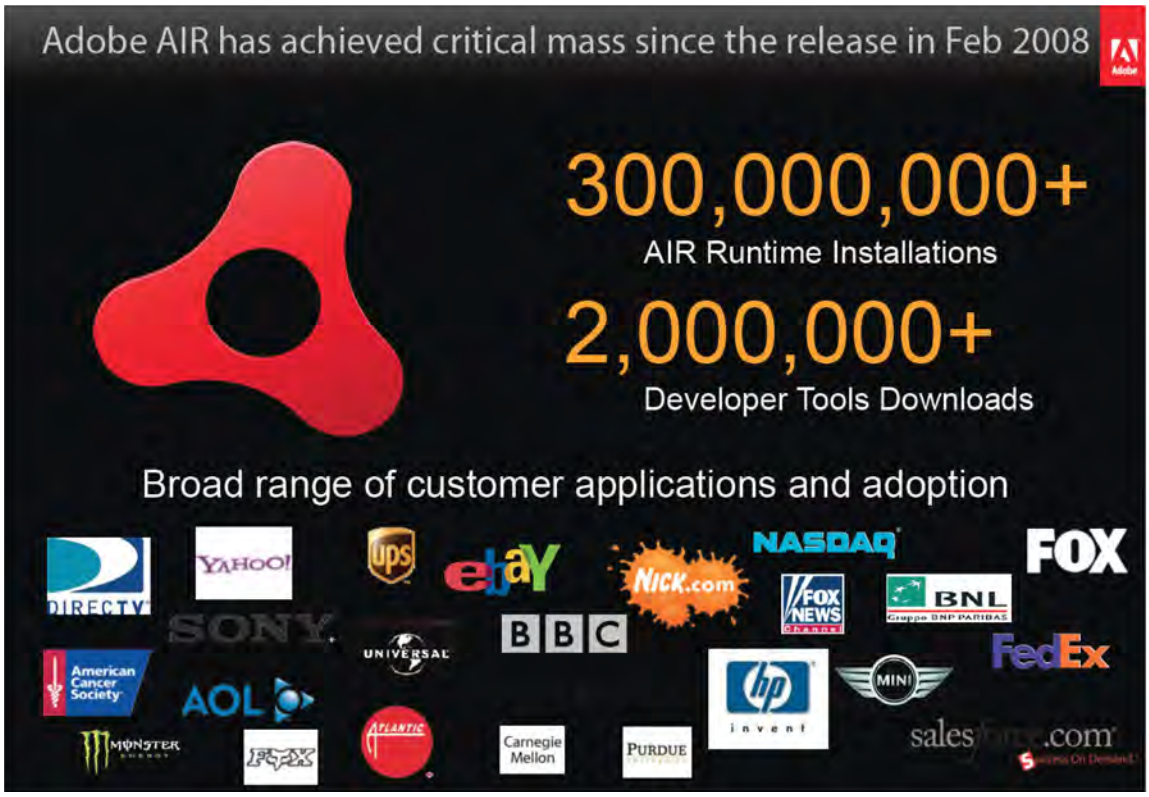

Figure 37: Adobe's AIR runtime has been rapidly making inroads within developer and user communities (image from [2]).

AIR is of course a pre-requisite for running AIR applications. If users do not already have the runtime when they try to install an AIR application, it will automatically be downloaded and installed (once) as part of the AIR application installation process. There are specific versions of AIR for Windows, Mac, and Linux, and each is typically an order of magnitude larger than the Flash browser plug-in. Unlike web applications which are simply loaded on-the-fly, each AIR application must be downloaded and installed in the same way as any other native desktop program. Once installed, AIR based programs are no longer constrained to many of the limitations imposed by web browsers and inherit many new desktop-specific capabilities such as offline operation, direct access to the user's local file system, and the ability to generate desktop notifications. Table 1 compares desktop and web applications on a per feature basis.

Table 1: Comparison of Web and Desktop Apps[52]

Feature	RIAs in the browser	RIAs on the desktop
Application delivery	Applications can be easily discovered, explored, and used.	Installed applications have more persistence, power, and functionality.
Installation	No application installation is necessary.	Applications install seamlessly from the browser or download and install like a traditional desktop application.
Application updates	Applications are updated by pushing new content to a website.	AIR provides APIs that allow applications to be updated as easily as pushing new content to a website.
Multiple OS support	Applications run on multiple operating systems and browsers.	AIR applications are cross-platform, so they can be installed on and run on multiple operating systems.
Programming languages	JavaScript is provided by browsers and ActionScript™ is provided by Adobe Flash® Player.	Integrated JavaScript and ActionScript virtual machines are compatible with the browser.
Background capability	RIAs can run only in a visible browser window.	Applications can run in the background or provide notifications like traditional desktop applications.
Persistence	Activity is limited to the browser session. When the browser is closed, information is lost.	RIAs are installed and available on the desktop. They store information locally and operate offline.
Desktop integration	Applications are sandboxed, so desktop integration is limited.	Applications can access a desktop file system, clipboard, drag and drop events, system tray/notifications, and more.
User interface control	RIAs run within a browser window that has its own controls, branding, and integration with the desktop.	RIAs have a customizable user interface and desktop integration, enabling branded experiences.
Data storage	Applications have limited local storage, which the browser can destroy.	Applications have unlimited local storage and access to a local database, plus encrypted local storage.

Interestingly, AIR based applications for social networking sites like Twitter and Facebook are particularly popular with their ability to (actively) provide features like background notifications and drag-n-drop file uploading.

A number of critics suggest AIR is a stopgap platform for pure web based applications. They go on to suggest most (if not all) of the functionality of the most popular AIR applications could be done using purely web-based applications. However, the success of applications like TweetDeck (Section 3.1), the NY Times Reader 2.0 (Section 2.2.6), and a growing number of AIR showcase applications[53] seems to weaken some of this criticism.

The NY Times Reader 2.0 application had an extensive feature wish list for version 2.0 of their online/offline news application. Using AIR, developers were able to provide a publication solution with support for real-time updates, notifications, social networking, multiple hardware devices, multimedia, and customizations (Figure 38).

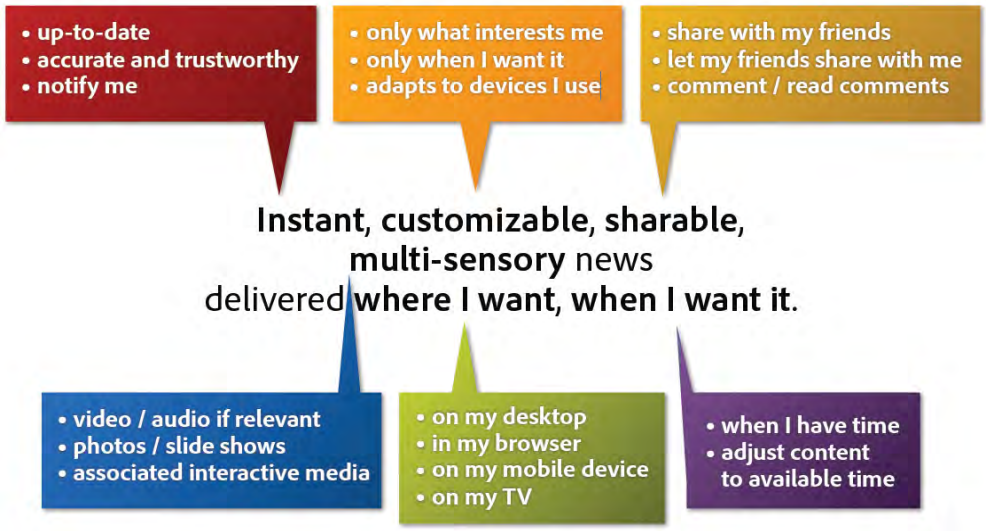

Figure 38: The New York Times had a tall order of expectations for their news reader application which they were able to fulfil using Adobe AIR[38].

2.6 Flash and PDF

Portable document format (pdf) is the world's standard file format for electronic documents, bar none. According to Adobe there are currently an estimated 200 million pdf documents in existence, and over half a billion people have used pdf documents[54]. Although Adobe invented the pdf format, it released pdf as an open ISO standard in July of 2008. The free Adobe Reader program remains the defacto standard for viewing pdf documents, while its big brother Acrobat (purchasable from Adobe) has additional features, in particular the ability to create pdf documents.

So aside from being attributable to Adobe, exactly what does pdf have to do with the focus of this paper, the Flash platform? The entire Flash player (version 9.0) has been integrated into Adobe Reader (and Acrobat) as of version 9.0 onward. This means Flash content or applications written for Flash can be embedded within a pdf document. No longer are documents constrained to text and static images only; it is now literally possible to include audio, video, and complete working applications within a 'document'. In fact, Annex A of this pdf document includes interactive samples of Flash programs and content which the reader can experience firsthand. This enhanced capability for embedding dynamic, interactive, multi-media rich applications within a pdf represents a significant shift in the paradigm of what exactly a document is and how information can be shared. Indeed, as with many Flash related technologies, the lines become blurred between what is content, document, and application.

Adding Flash content to a pdf document is quite easy using Acrobat (Figure 39). Begin by selecting 'Flash Tool' from the 'Multimedia' menu bar. Next select the Flash target area within the document, and finally select the Flash (swf) file to insert. For more detailed information regarding embedding Flash content using Adobe Acrobat including security and other specific limitations, refer to Adobe's "Security for Flash Player compatible content in Acrobat 9" technical white paper[55].

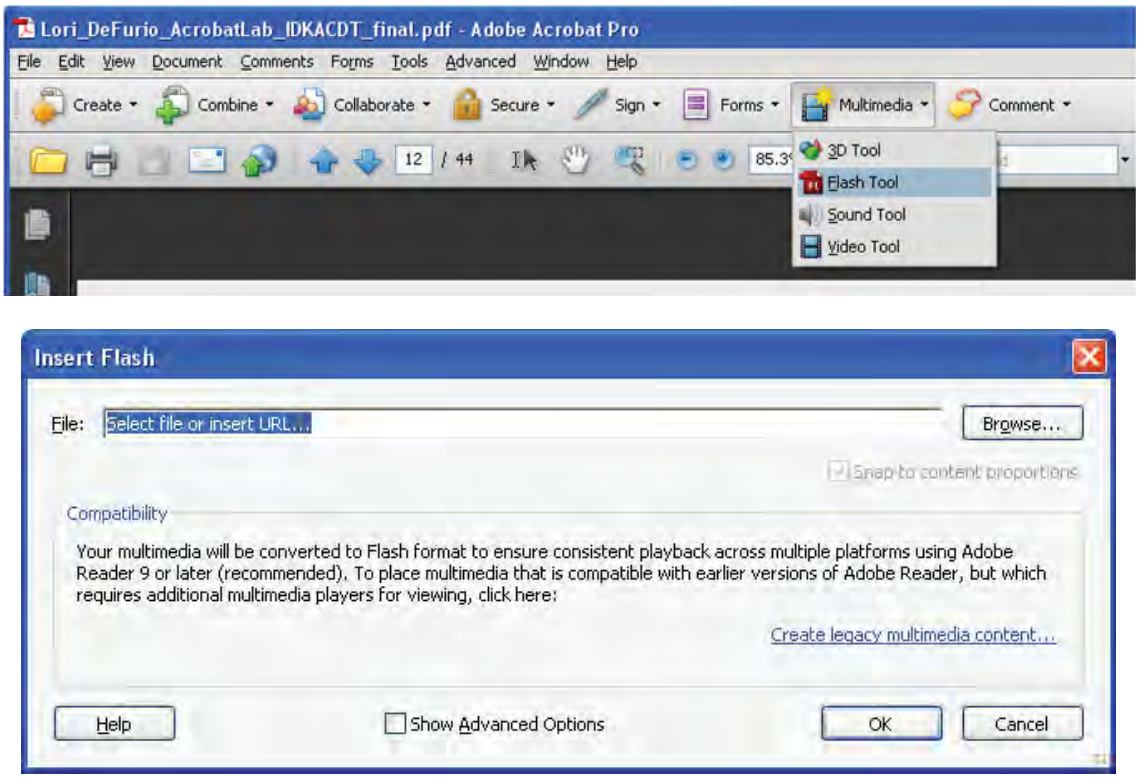

Figure 39: Inserting Flash content into a pdf document using Acrobat's 'Flash Tool'.

James Ward's Flex Dashboard application[56] is an example of a Flash application (RIA) inside a pdf document. In fact the online web version of the Dashboard application contains a button to generate a pdf version of itself.

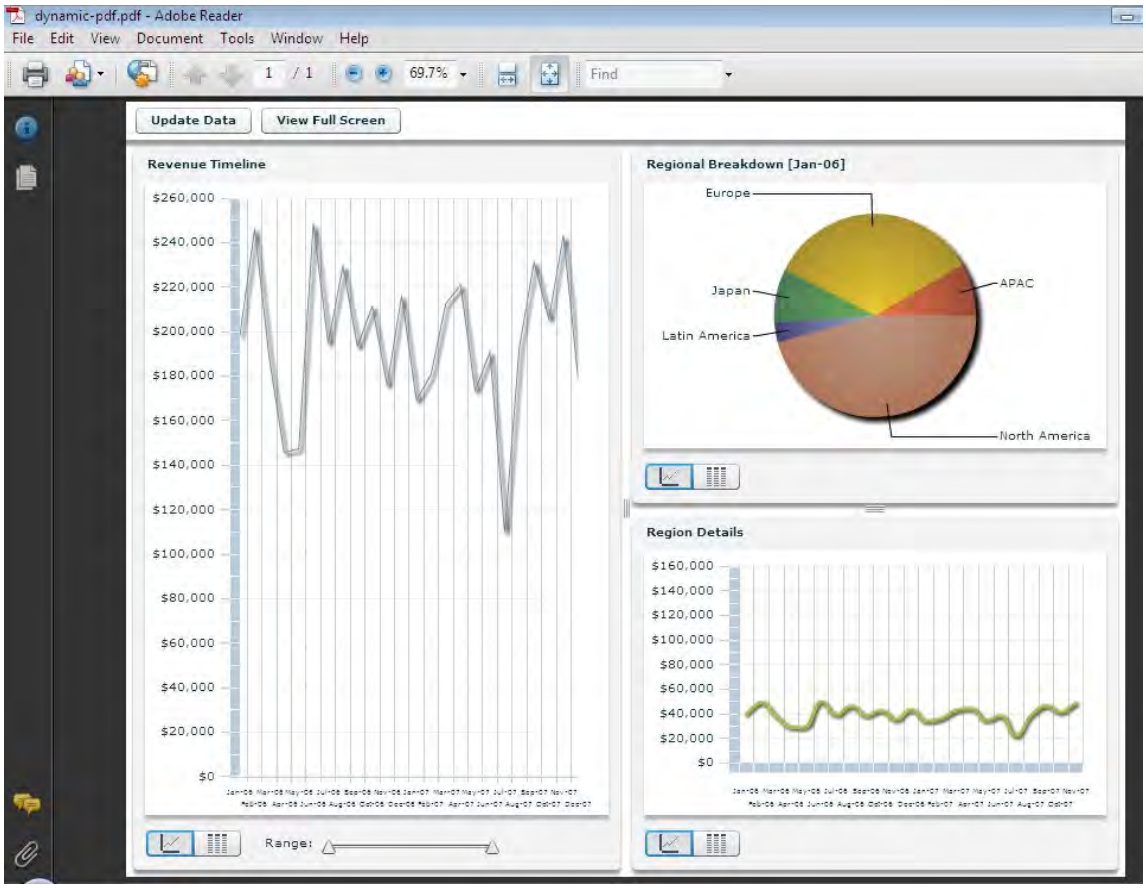

Figure 40: A Flash based interactive data visualization (Flex Dashboard) [56] embedded within a PDF document executes seamlessly.

Often there are several means to convey information and depending on the context and nature of the topic, certain methods will be more effective than others. In other words, while sometimes a picture is worth a thousand words, depending on the circumstances, so too is an audio recording, video message, dynamic visualization (Figure 40), and other multimedia forms. For example many scientific papers contain large amounts of complex data which could benefit from advanced interactive and dynamic visualizations (Section 3.7).

As mentioned, many MICS research projects manifest (at least in part) as concepts for new or modified user interfaces. Exploring, refining, and implementing these design proposals often necessitates an iterative approach in which input is solicited from subject matter experts (SMEs). Having a tangible interactive display with dynamic and operationally relevant context for SMEs to evaluate and subsequently provide feedback can greatly enhance and expedite the design process. While directly distributing prototype software applications themselves is one means for sharing proposed designs with SMEs, pdf documents provide another universally accessible format for distributing and experiencing applications that transcends firewalls while simplifying and minimizing local host requirements for execution.

Most of the research output from DRDC is published as (pdf) reports, which are often released with a corresponding live presentation. However, unlike the pdf, the presentation is typically a one-time event for a small local audience. Imagine the benefit of having a video of the author's presentation permanently attached as an appendix to the published pdf research paper; anyone with access to the pdf would also have the benefit of watching the author's presentation.

One final note regarding Flash and pdf: not only can Flash content be embedded inside a pdf, it is also possible to both generate and view pdf files inside of the Flash player. AlivePDF is an open source library for generating pdf files from within a Flash client. Using AlivePDF, it is possible to create new pdf pages and add text, graphics, and images all from within Flash. FlexPaper[57] is a lightweight open source (client-side) component created by Erik Devaldi for viewing, searching, and printing pdf files from within the Flash player (Figure 41). Adobe's LiveCycle ES enterprise software (Section 4.2.5) provides similar Flash based pdf capabilities using server-side technologies.

Figure 41: Pdf documents can be viewed directly inside the Flash player using FlexPaper[57] (no pdf reader required).

2.7 Ubiquity

If there is one single greatest advantage of the Flash platform in general, it would probably be its ubiquitous nature. The browser plug-in is far and away the most common instance of the Flash player. Currently at version 10, the Flash browser plug-in is available for all major web browsers on all major OS platforms (Table 2).

Table 2: Flash web browser plug-in availability by OS platform[58].

Platform	Browser	Player Version
Windows	Internet Explorer (and other browsers that support Internet Explorer ActiveX controls and plug-ins)	10.0.22.87
Windows	Firefox, Mozilla, Netscape, Opera (and other plugin-based browsers)	10.0.22.87
Mac OS X	Firefox, Opera, Safari	10.0.22.87
Linux	Mozilla, Firefox, SeaMonkey	10.0.22.87
Solaris	Mozilla	10.0.22.87

Impressively, the Flash player can be found on over 98% of the world's computers with an internet connection[3] (Figure 42). In comparison, the Flash player's biggest direct competitor, Microsoft Silverlight, currently sits at about half the Flash player's penetration.

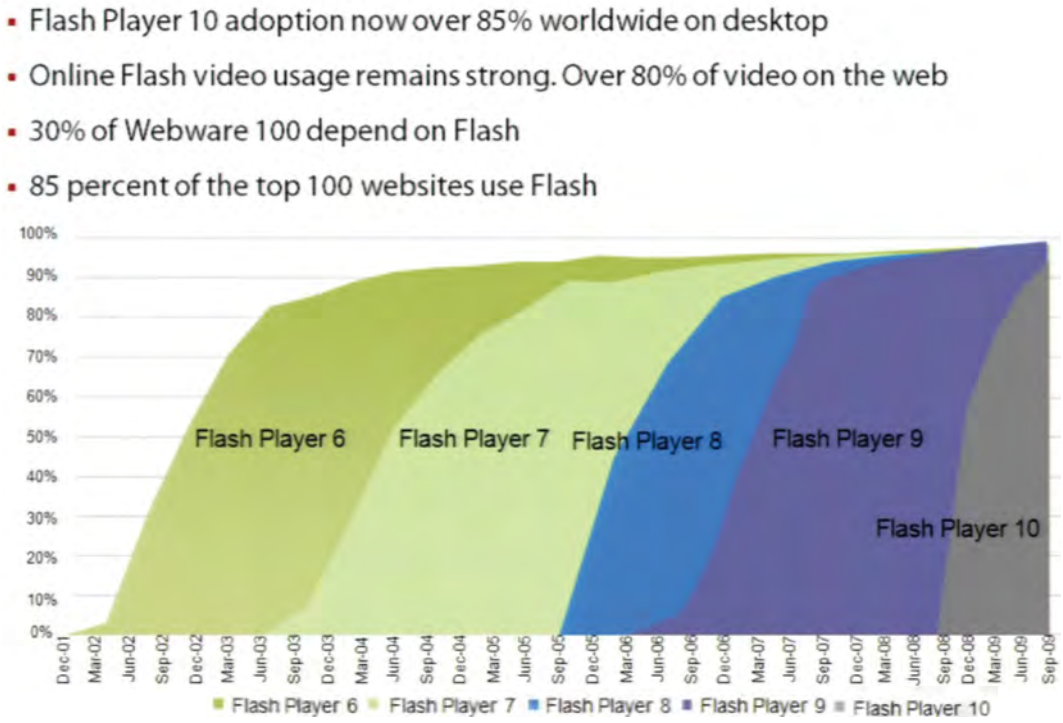

Figure 42: Flash player penetration and adoption rates have been consistently very high (Image from [1]).

While the most common means for experiencing Flash content may be inside a browser, it is certainly not the only option. Adobe has expanded the Flash player presence beyond its browser plug-in roots to many new areas (Figure 43). For example, Adobe's AIR runtime environment (Section 2.5) for desktop applications includes the full Flash player. AIR's Flash player is entirely separate from the Flash browser plug-in, for example users can experience AIR applications without even having the Flash browser plug-in.

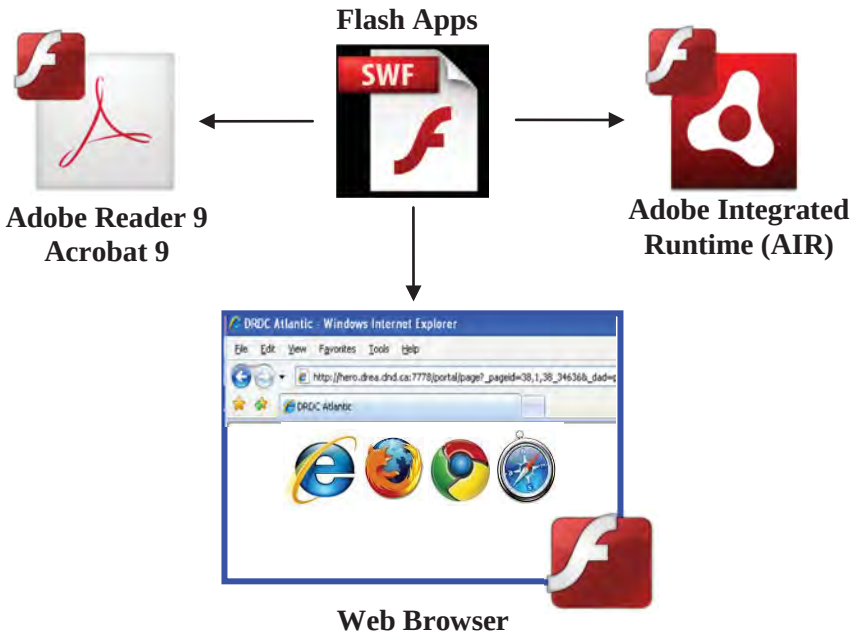

Figure 43: The Flash player inside of web browsers, Adobe Reader/Acrobat, and the Adobe Integrated Runtime provides 3 wide reaching options for playing back Flash applications and content.

As mentioned, Adobe has also integrated the full Flash player into Adobe Reader and Acrobat as of version 9. This means Flash content and applications can be experienced within a pdf document. The Mortgage Calculator[59] (Figure 44) is another simple example of ubiquity using Flash; not only is it available as an online web application and desktop installed widget, it is also available embedded within a PDF document. The Mortgage Calculator is a great candidate for a portable (pdf) application; it is small, completely self contained, and doesn't leverage any device or platform specific capabilities. The Mortgage Calculator application, along with several other examples of Flash embedded inside of a pdf document, is included in Annex A this document.

Figure 44: The Mortgage Calculator[59] is available in separate web, desktop, and pdf versions.

Not satisfied with 'just' the web and desktop client segments, Adobe have been aggressively pushing the Flash player to the mobile phone market as well. In the past this has been accomplished using a lightweight version of the Flash player - Flash Lite (Section 2.4). Despite not wanting to fragment the Flash platform, a lighter version of Flash was necessary to accommodate the very limited hardware resources associated with many of these handheld devices. More recently however Adobe have begun efforts to re-unify to a single, common core player – "One Build to Rule Them All!"[60]. In support of this goal, and beginning with version 10.1, Adobe have made a concerted effort to optimize the Flash player for mobile internet devices such as net-tops, netbooks, and smartphones. These optimization efforts target things like faster rendering, lower memory consumption, and less battery drain. Note the 'common core' is about 80%, as there will always be a portion of the Flash player which is custom or unique to the device's capabilities (e.g., GPS, multi-touch, accelerometer).

Whether it is the lightweight or full version, the Flash player is currently available for Google's Android, Nokia's Symbian OS, Windows Mobile, and the Palm Web OS. In fact Adobe has partnered with 19 of the world's top 20 mobile device manufacturers to ensure a Flash presence. This list covers all the major smartphone players except for one glaring omission, namely Apple's iPhone.

The issue between Apple and Adobe regarding adopting the Flash player on iPhone OS devices has been the subject of much speculation, rumours and discussion[61]. As far as global smartphone market shares are concerned, Apple currently sits third at about 18%, just behind RIM's 20% , and very far back from the Global leader Nokia's 40% share[62]. Given that the Flash player has been successfully demonstrated on the vast majority of competing smartphones, it would appear there are no technical reasons to justify its absence from the iPhone. However, the iPhone is not a complete shutout to the Flash platform, as Adobe have announced the upcoming versions of Flash Professional and Flex Builder have the ability to compile Flash applications into native iPhone format. In fact at the Max 2009 keynote, Adobe demoed several Flash built iPhone applications on the iPhone. This ability to compile ActionScript based code into native applications for iPhone represents a significant deviation from Adobe's 'write-once-run-anywhere' (WORA) paradigm to a 'write-once-compile-anywhere' (WOCA) approach.

It is not hard to imagine the motivation behind the interest in mobile platforms, as both the capabilities and popularity of mobile phones worldwide continue to enjoy explosive growth. Smartphone sales currently rival notebooks, and market analysts estimate worldwide sales of smartphones will surpass PCs within 2 years[63-64]. Generally speaking there is a definitive trend in computing platforms towards smaller, more mobile and increasingly connected devices, and Flash's mobile penetration is well on its way to matching its desktop dominance.

In addition to desktop PCs and mobile devices, Adobe has been working with companies like Intel[65] to extend Flash on to the 'third' screen – 'connected' living room devices like digital televisions, blu-ray players, and set-top boxes. Flash on these connected devices can be used to deliver internet video (e.g., Hulu), live information feeds, notifications, social networking and other interactive content.

The Open Screen Project (OSP)[64] is an Adobe-led initiative with multiple industry partners aimed at positioning the Flash player as 'the' common runtime for delivering content and applications across all categories of information devices. In Adobe marketing speak the goal of

the project is to "Enable consumers to engage with rich Internet experiences seamlessly across any device, anywhere"[66]. The current list of close to 50 partners reads like a who's-who of the high-tech industry with names like Motorola, Nvidia, Google, Cisco, Intel, Sony, Fox, Disney, Nokia, RIM, and the list goes on. All these companies have basically agreed on Flash as a standard client-side runtime; each will incorporate and support the Flash player and Flash content on their respective platforms and services. Adobe for its part has agreed to several concessions including abolishing licensing fees for the Flash player and AIR, removing use restrictions on the swf and flv file formats, as well as publishing Flash specific APIs and protocols related to client-server communication.

Figure 45: The Adobe led 'Open Screen Project' is an industry-wide initiative that aims to "Enable consumers to engage with rich Internet experiences seamlessly across any device, anywhere"[66].

Ian Williams speaking from the 2009 Intel Developer Forum in San Francisco summarized Adobe's plan for world domination quite succinctly:

> *"Adobe is hoping to implement Flash in just about every PC, mobile, TV and consumer electronic device on the planet, allowing developers to create a range of widgets and applications that run across all of these devices"[67].*

The ubiquity of Flash, or any runtime environment, is important for a number of reasons. From an end user perspective, it equates to more applications, media content, and services which are accessible and compatible with more devices, from more places, and at more times (e.g. whether online or offline). For developers, a common platform simply means developing, deploying, and supporting one version of an application as opposed to many.

Finetune is an early example of an application which takes advantage of the ubiquity of the Flash platform. Finetune is an online social music site which enables customers to discover, listen, share, and manage playlist-based music. Thanks to Flash's ubiquity, Finetune customers can

access and use Finetune via separate web, desktop, tv, mobile, facebook, and Wii versions. Despite the diversity of hardware and access options, developers leverage the same Flash tooling and source files across a wide spectrum of consumer devices.

Figure 46: The social music sharing application Finetune[68] exemplifies the ubiquity of Flash having instances for gaming consoles, browsers, desktops, smartphones, TVs, and social networking platforms.

2.8 Flash Media Server (FMS)

Not all aspects of the Flash platform are directed to the client side of application architecture. Due to the relatively large file sizes involved with video, application development must often extend beyond the client and into the server realm. Adobe offers a suite of video server tools, most notably Flash Media Server. Flash Media Server (FMS) is essentially a set of server-side tools that enable video content to be streamed out to Flash player clients. FMS can be used for:

- *Video on Demand, streaming video stored on the server to the flash client.*
- *Live Video, a server-side application* that *allows users to broadcast their own video from a webcam on website with live stream Flash video player to other users or to the server for recording and on demand viewing later.*
- *Real Time Communication, an application which requires collaboration between multiple clients, such as a chat room or multiplayer game.[69]*

Two noteworthy and popular alternatives to FMS are Red5 Flash Server[70] and Wowza Media Server Pro[71]. Red5 is an open source Flash server that supports *"audio/video streaming (FLV, h264, AAC, and MP3), Recording Client Streams (FLV only),Shared Objects, Live Stream Publishing, and Remoting (AMF)"*[70]. Wowza Media Server Pro is a commercial product available in various editions which the company describes as "a high-performance, extensible and a fully interactive Flash media server for live and on-demand streaming, chat, recording and much more"[71].

AskMeFlash.com[72] provides a handy summarized feature comparison (Table 3) between the three competing Flash media server solutions.

Table 3: Feature Comparison between Adobe, Wowza, and Red5 Media Servers[72].

Feature	Flash Media Server(FMIS 3.5)	Wowza Pro Unlimited with MPEG- TS	Red5
Protocols supported	RTMP RTMPT RTMPS RTMPE RTMPTE RTMFP	RTMP RTMPT RTMPS RTMPE RTMPTE	RTMP RTMPT RTMPS
Developer edition	10 Connections (Free)	10 Connections (Free)	Free
Pricing	$4500	**$995**	Free(Open Source)
Supported Platforms	Microsoft® Windows Server® 2003 with Service Pack 2 or Windows Server 2008 Linux® Red Hat® 4 or 5.2 Runs as a 32-bit software on both 32- and 64-bit operating systems.	Windows Mac OS X Linux Solaris Unix 64-bit Support on all	Windows Debian/Ubuntu Mac OSX WAR Gentoo
Video Streaming (live and on-demand)	FLV H.264	FLV H.264	FLV F4V MP4
Audio Streaming	FLV MP3 HE-AAC	FLV MP3 HE-AAC	MP3 F4A M4A
Recording	H.264/AAC to FLV container	H.264/AAC to FLV container H.264/AAC to MP4 (Quicktime) container	FLV Only
Action Method Format 3 (AMF3)	AMF3	AMF3	AMF
Server Side	AS2	Java	Java

Regardless of the back-end server solutions, for larger scale video providers content delivery (or distribution) networks (CDN's) become increasingly important. CDN is a strategy for balancing server and bandwidth demands which involves hosting copies of data (e.g., videos) on numerous computers spread across a network to avoid bottlenecking a single server and/or its network connection.

Whether live, pre-recorded, professional, or amateur, video streams are being incorporated into more and more of the web sites, services, and applications people experience everyday. Examples are virtually everywhere including news, social networking, peer-to-peer, and entertainment sites. CNN Live with Facebook[73] is a mashup of live news video and social networking. For example, while watching CNN.com's live video stream of President Obama's inauguration, Facebook users were able to post live status updates and see the status updates of both friends and everyone else watching (Figure 47).

Figure 47: A facebook community of friends can privately share a streamed video broadcast as a group in real-time[73].

Ustream.TV (Figure 48) describes itself as the "live interactive video broadcast platform that enables anyone with a camera and an Internet connection to quickly and easily broadcast to a global audience of unlimited size" [74]. The real-time video sharing service relies heavily on Flash server side video technologies, or more specifically Adobe's Flash Media Interactive Server product. The site claims unique experiences over the mountains of pre-recorded video on the web including live interactive events such as:

- *Major political events such as debates, speeches, rallies*
- *Talk shows*
- *Entertainment events such as premieres and 'red carpet events'*
- *Music showcases of their favorite music, of their own band's performances, and live jam sessions*
- *Conference sessions*

- *School and business events and training*
- *Sporting events at college and high school level*
- *Personal milestones such as holiday gatherings, weddings, grade school events, parties, even births*
- *Interactive games for viewers to watch or join*[74].

Figure 48: A live video broadcast using Ustream[74] and Flash based technology over the internet.

2.9 Flash Data Services – LiveCycle DS and BlazeDS

Flash applications typically work with data, and while this data may be stored on the user's local machine, more often than not it is located on a data server somewhere else on the network. Flash provides an assortment of options for Flash clients to connect and exchange data with most server types including PHP, Ruby, Java, Cold Fusion, .NET, and others.

Flash clients have long had the ability for basic text and xml data exchanges with back-end servers using hypertext transfer protocol (http), the same protocol the web browser uses to retrieve html documents. This is the simplest means of establishing data communication, as there is no requirement of any Flash-specific services on the server side. While simple to implement, text based communication can be relatively verbose, slow, and generally does not scale well. For example, xml data being retrieved from a server has to be serialized, deserialized, and parsed by the Flash client.

For Flash applications that are heavily data-driven and require an improved set of connectivity options for client-server communication, Adobe offers a family of server-side data service products. 'BlazeDS' is Adobe's open source data service solution; a set of server-side Java remoting and messaging technologies which provide enhanced and added options for Flash client-server communication. Adobe also offers a suite of commercial data service products known as 'LiveCycle DS'. Differentiating between these data service solutions and their respective features is notoriously confusing (Table 4). They are first distinguished by whether or not subscription based technical support is available. Capability wise they are basically differentiated by whether or not they support data management and advanced messaging.

Table 4: Feature comparison between BlazeDS and LiveCycle DS[75].

Feature	Blaze DS	LiveCycle DS Community Edition	LiveCycle DS Single CPU	LiveCycle DS
Cost	Free	Free	Free	Paid
Support Available	No	Yes	No	Yes
License	Open Source	Open Source	Commercial	Commercial
AMF Communication Channels	Yes	Yes	Yes	Yes
Java RPC Services	Yes	Yes	Yes	Yes
Messaging	Yes	Yes	Yes	Yes
Clustered messaging	Yes	Yes	Yes	Yes
Pub/sub messaging	Yes	Yes	Yes	Yes
Servlet based messaging	Yes	Yes	Yes	Yes
NIO based messaging	No	No	Yes	Yes
Data Management	No	No	Yes	Yes

All these products provide several significant capabilities for improving communication between Flash client applications and their servers. For example all these data service solutions support Action Message Format (AMF), a binary data format (as opposed to ASCII) which enables very fast and efficient data exchanges between Flash clients and servers. Java remote procedure calls (rpc) allow a Flash client application to directly invoke methods of java programs running on the server. The messaging services enable publish and subscribe capabilities and real-time data exchanges (as required for collaborative applications).

The Collaborative Forms application (Figure 49) is a simple example which uses BlazeDS to provide real-time data messaging. "Users in different locations can fill in forms "together" in a

real-time and in-context collaboration session: changes made by one user are automatically reflected in the other user's application”[76].

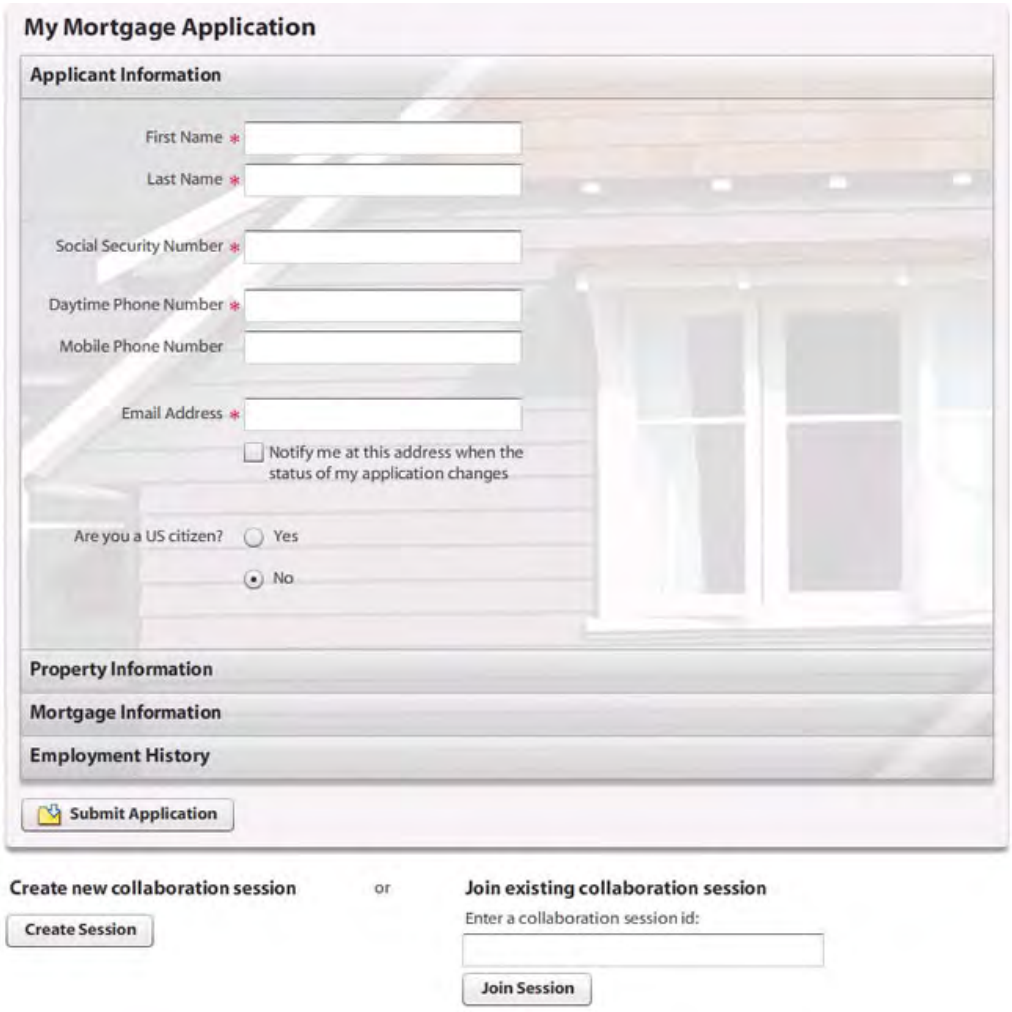

Figure 49: BlazeDS handles all the real-time data messaging for this Collaborative Forms application[76].

3 Extended Capabilities

The following section discusses a selection of extended capabilities of the Flash platform, focusing on areas which are more interesting from a military research and experimentation tool.

3.1 Social Networking

In the context of the web, social networking generally refers to the use of websites or services to find and stay connected with family, friends, colleagues, or people with similar interests. Sharing within these online communities is by definition more social oriented; people often share things like praise, sympathy, opinions, advice, jokes, and life experiences. Communication can take many forms including email, messaging, tweets, blogs, forum postings, chat, etc.

Wikipedia[77] maintains a growing list of currently well over a hundred social networking sites, and the list continues to grow. Currently some of the more popular sites include myspace, digg, twitter, and the 800lb social networking gorilla known as facebook. Flash has emerged as a preferred platform for developing 'socially aware' applications, and Flash specific APIs exist for all these sites and many more. For example, a quick search on Google code using the social networking website name and "flash" or "as3" as keywords will provide a list of Flash based APIs for these social networking websites.

Facebook proudly publishes usage statistics on their website, and admittedly some of the numbers are impressive. For example, as of this writing there are well over 300 million people actively using facebook and over 350,000 applications available on facebook. A very large number of these apps, including the majority of the most popular apps, are Flash based[79]. In fact, when facebook developers decided to create their own desktop program for facebook, they did so using Adobe AIR (Figure 50).

Figure 50: Facebook's creators chose Flash to develop their desktop version of facebook[78].

Developers have in fact been building facebook applications since the platform's launch, long before Adobe and facebook's formerly announced partnership in early 2009[79]. Adobe has certainly recognized the significance of facebook, having a separate and dedicated facebook sub-section[80] on the Adobe developer area of their website. The facebook section contains APIs, sample code, tutorials, links, and other developer resource tools to aid in creating Flash applications for facebook.

Twitter[81] is a social networking web site and service through which users can send and read small text messages (aka ‘tweets’) in real-time. “Tweets are text-based posts of up to 140 characters displayed on the author's profile page and delivered to the author's subscribers who are known as followers.”[82] “More than half of Twitter users (55%) use something other than Twitter.com”[83].

A popular example among twitter enabled apps is an Adobe AIR application called TweetDeck[84]. TweetDeck enables users to immediately receive and send tweets from their desktop (or mobile) without opening a browser, navigating to the twitter web site, and logging in. TweetDeck has been steadily expanding its social networking capabilities and has recently added integration with both facebook and myspace. Using TweetDeck, users can stay organized and up to date with all their friends and colleagues, regardless of the social networking platform (Figure 51).

Figure 51: Not satisfied with being the most popular app for twitter, TweetDeck now integrates with facebook and myspace, enabling users to easily monitor and update all 3 popular social networking sites (image from TweetDeck.com [84]).

3.2 Collaboration

Within the context of this paper, collaboration basically refers to real-time communication using computer devices over a digital network. This can include live (text) chat, voice, video, document sharing, white-boarding, and other desktop sharing technologies – all of which also happen to be common ingredients to the Flash platform. Like most applications in general, the more 'rich' these collaboration clients become, the more apt they are to favour Flash construction over alternatives such as java, HTML, or .net. Again rich in this context suggests client side applications that incorporate more graphics, audio, video, animation, and/or interactivity. Because there are multiple clients involved, collaboration inherently requires some form of network architecture through which the various clients can communicate. Options include peer-to-peer (in which no server is involved), or more often some type of client-server architecture in which all communication is routed through a central server. Flash server side technologies like BlazeDS for data and Red5 for video, combined with highly optimized messaging protocols like AMF continue to ease and enhance the development and deployment of real-time collaborative applications.

Flash based collaboration software is often available in both browser and desktop versions. As with many other Flash applications, they are also beginning to appear more and more on smartphone devices. Based on the continued rapid growth in both capability and popularity of handheld devices, combined with Adobe's ubiquitous positioning of Flash across the entire market, expect this trend to continue even more so.

The Map Rooms demo (Figure 52) created by Adobe Flash developer Christopher Coenraets is a simple yet great example of a Flash based collaboration application. In addition to chat messaging, users share a map that they can graphically annotate in real-time. "Map Rooms works like Chat Rooms. You can create a room, or join an existing one. In addition to chatting, Map Rooms allows you to collaborate on a map: the application leverages the real time capabilities of BlazeDS or LCDS to provide map synchronization between users in the room, and allows you to "whiteboard" on top of a map"[85].

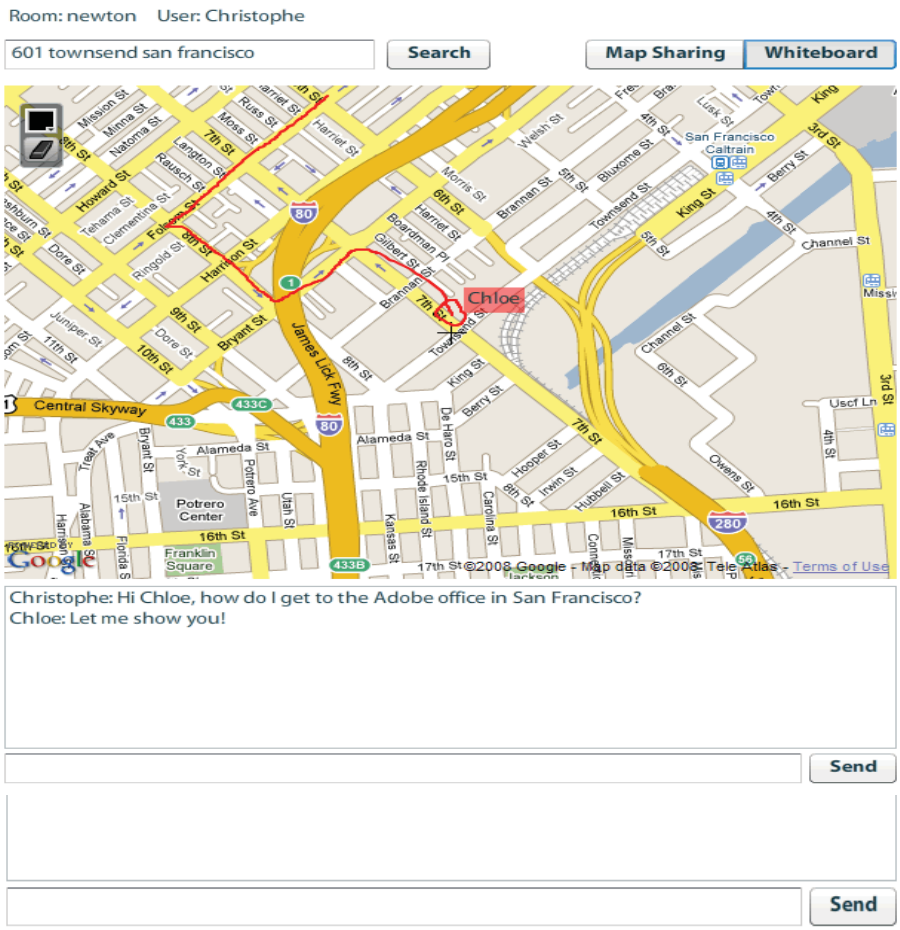

Figure 52: The Map Rooms[85] demo uses the Google Maps AS3 API to enable users to share a map, chat, and annotate in real-time.

When it comes to collaboration technology, nowhere is there more hype than with the recently introduced Google Wave[86] platform. "Google Wave is a real-time collaborative tool that permits groups to interactively 'converse' on a project, using richly formatted text, photos, videos and maps. Real-time here is taken to the extreme, with key-stokes shared among participants as they occur--no waiting for the press of a return key to send your thoughts along."[87]. Google defines a 'Wave' as "both a conversation and a document where people can discuss and work together"[86]. As often happens with new technology, Google Wave tends to blur the lines between informational and technological categories, in this case merging application, document and media types.

As of this writing Google Wave is in limited beta testing and still very much a technology in its infancy. Nonetheless, it is an open source project and Google has started to release Wave APIs for other relevant development platforms including Flash. The Google Wave Flash APIs enable developers to create Flash clients which incorporate Google Wave collaborative technologies.

An early adopter of both Flash and Wave technologies is Ribbit, a company specializing in providing an enterprise level Flash based VOIP and messaging SDK for developers. “Ribbit is a fully programmable communications platform that allows web developers to integrate telephony and messaging capabilities into any workflow or application.”[88]

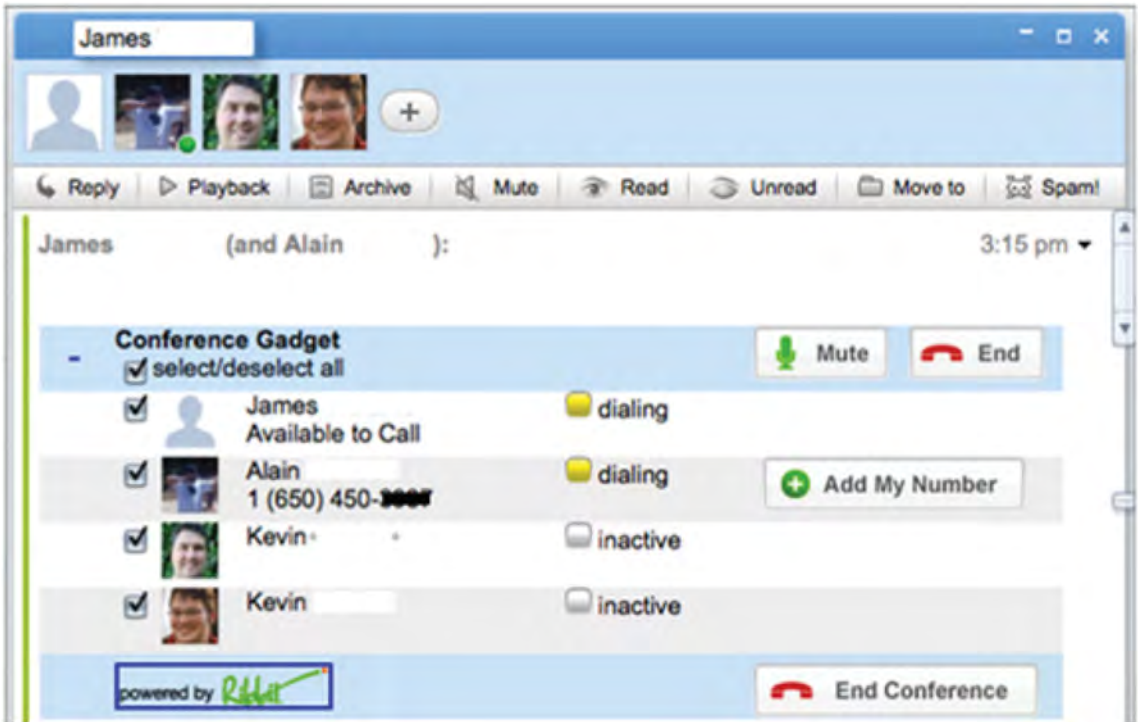

Figure 53: The Ribbit Conference Gadget[89] allows Google Wave participants to talk with each other while collaborating in real-time.

Ribbit showcases numerous VOIP enabled applications on their web site. One such example is the Ribbit Conference Gadget (Figure 53):

> *The Ribbit Conference Gadget allows (Google) Wave participants to escalate an online collaboration session to a real-time audio communications session, allowing participants to talk with each other while collaborating. The Ribbit Conference Gadget is persistent in the Wave and allows any Wave participant to:*
>
> - *Create an audio connection with multiple Wave participants*
> - *Add non-Wave participants to the session*
> - *Mute or hold any of the individual participants from the stream*
> - *Disconnect any participants from the stream*
> - *End the session[89].*

Another Google application which falls under the collaboration category is R.stoeber Group's Google Voice Desktop (Figure 54) application. Google Voice is a free service provided by Google for managing phone calls, voice mail, and short message service (SMS) text messages. Users of the service gain a new phone number which they can then use to route incoming calls to their existing home, office, and mobile numbers based on criteria such as the caller identity and the user's current location. Other features include online voicemail access, automatic voicemail transcriptions, conference calling, and to top it off calling is free within the US and Canada. Currently the service remains in beta and available only to US citizens. The Google Voice Desktop application provides quick and easy access to many Google Voice features, as well as enabling a more persistent means for voice and messaging notifications. Furthermore, being an AIR based application it runs on Windows, Mac, and Linux systems.

Figure 54: Google Voice Desktop [90] is an AIR application for accessing and managing many features offered by Google's free phone service.

For further examples of collaborative style applications built using Flash, readers may wish to visit the collaboration showcase section on Adobe's website.

3.3 Games

Flash technology has historically enjoyed a longstanding and integral relationship with the video game industry, having components that are widespread in both game development and deployment. Being one of the more common use cases for Flash, games are a significant driving force pushing both the creative and technological envelopes of the Flash platform. As end-users, gamers fuel ongoing improvements to the Flash player in areas such as stability, performance, and cross-platform compatibility. For developers, having a large and diverse game development community translates into more tools and resources such as APIs, components, tutorials, and code libraries, as well as a larger peer support base to share experiences and advice.

Generally, the role of Flash related technologies within the gaming industry differs significantly between the 'casual' (e.g., simple web-based) and 'enthusiast' (e.g., retail installed) game markets. For example, it is not uncommon for web-based games to be developed completely within a single tool such as Flash Professional. On the other hand, more complex and larger scale game productions (e.g., 3D first-person shooters) often employ several development tools and technologies, of which Flash may be just one constituent.

Despite being so lightweight in comparison to their native desktop brethren, casual games represent a significant and growing segment of the overall global gaming industry. Casual games (Figure 55) are typically a relatively simpler and smaller class of games (e.g., tetris), designed to load quickly and run inside a web browser. Due to the limitations of file size and performance inside a browser, these games tend to use simpler graphics (e.g., almost always 2D), relatively low resolutions and frame rates, and avoid complex visual and environmental effects. However, thanks to ongoing performance improvements in programming techniques, compilers, and runtime clients such as the Flash player, the fidelity of web based gaming continues to rapidly improve. This trend is further amplified by the ever progressing capabilities of computer hardware including display technologies, connection speeds, memory capacities, and processing power. More recently, casual games have been experiencing healthy growth within the exploding handheld device market. Again much of this trend is attributable to the increasing hardware and connectivity capabilities of mobile devices such as tablets, netbooks, and smartphones.

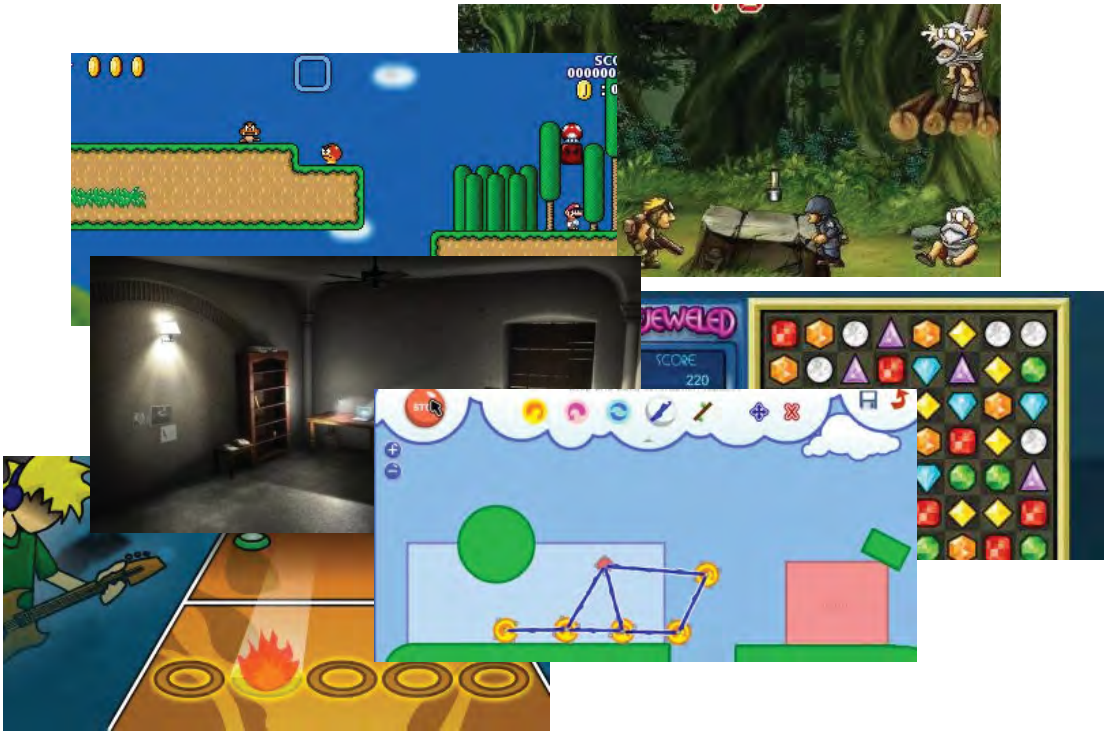

Figure 55: Game developers expand the boundaries of the Flash platform both artistically and technologically, particularly with respect to graphics, animation, and interactivity (images from techcult.com).

It is interesting to note the current trend towards more social and collaborative applications mentioned previously applies equally to the online gaming sector. There is an increasing fusion between gaming and social networking which is proving mutually beneficial for both market segments. For example, multi-player games are increasingly integrated with social networking websites like facebook and mySpace. Farmville (Figure 56-left) is a very popular example which integrates social networking features provided by facebook with a casual Flash based game.

Figure 56: Farmville (left) is a hybrid game and social networking Flash application played by tens of millions of people daily. League of Legends (right) uses Flash data server and messaging technologies to connect players from around the globe.

In contrast to casual style games, more complex enthusiast games (e.g., 3D first-person shooters) often employ several development tools and technologies. While perhaps not immediately associated with these larger gaming productions, Flash tooling is nonetheless quite prevalent, albeit within more specific development roles. League of Legends (Figure 56-right) is an example of a desktop installed 3D game which uses Adobe's LiveCycle data services (LCDS) to manage its online multi-player data exchange and messaging requirements. It also happens to use a Flash based interface for configuring, launching, and joining multi-player games. Despite the integration between League of Legends and the Flash technologies it does use, the game engine itself is not Flash, but rather a native OS application (presumably because of 3D performance requirements).

Also worth noting is Scaleform GFx (Figure 57), a commercial suite of tools used in the production of many popular retail games. This Flash-based third-party toolset can be used for creating numerous different game elements including menus, HUDs, videos, maps, lobbies, cut-scenes, and animated textures. Scaleform is completely cross-platform friendly with support for all major desktop, console, and handheld gaming platforms. As "the #1 video game UI solution"[91], it has been used in hundreds of high-profile retail games over the past few years, including games based on the Epic Unreal and Crytek CryEngine engines.

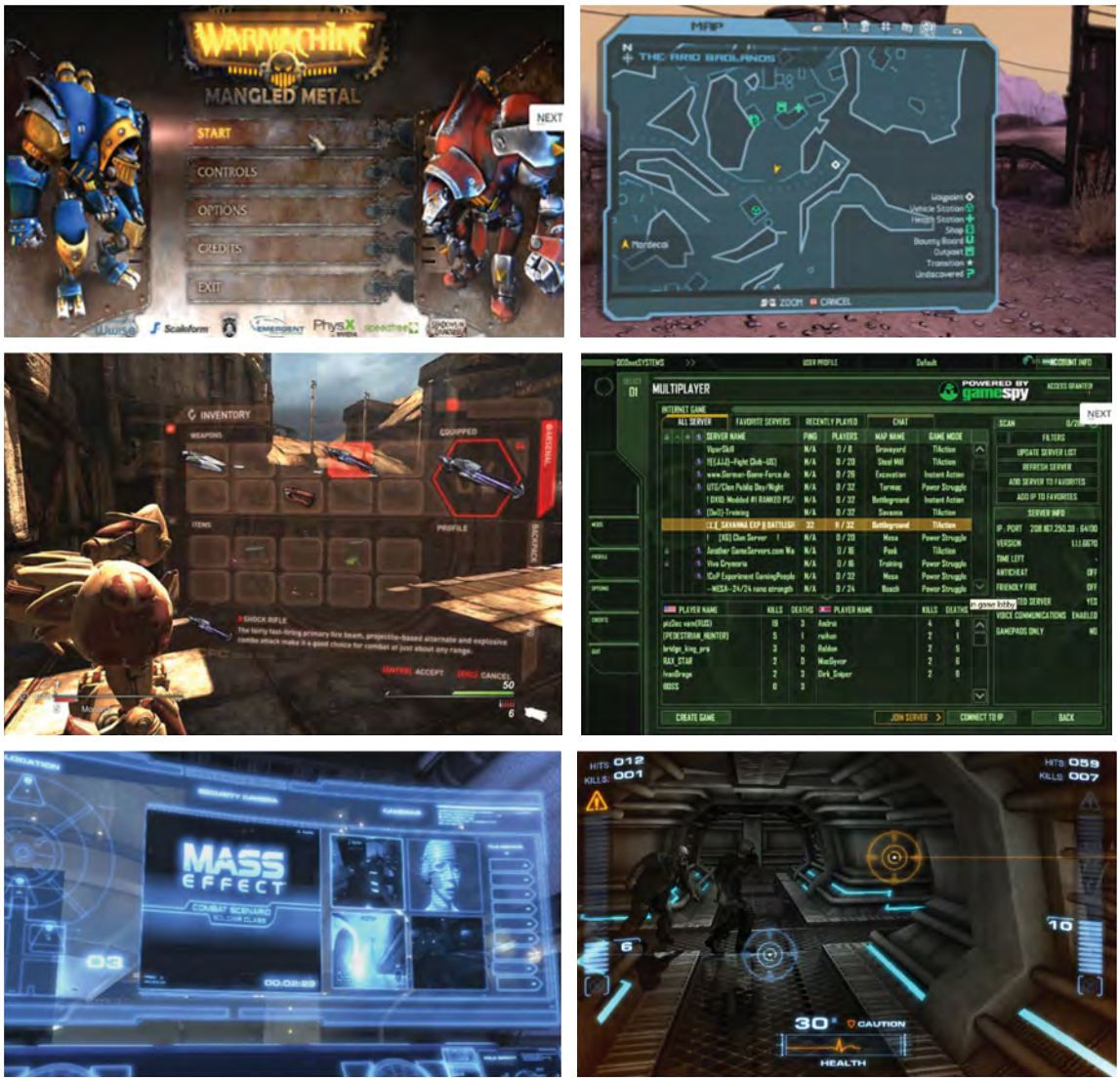

Figure 57: [Scaleform GFx][91] is a Flash-based toolset used to create and deliver several key elements within high-end game productions including (from top left by row): launch menus, maps, in-game menus, player lobbies, in-game videos, and HUDs.

3.4 Human Computer Interface (HCI) Research

3.4.1 Interface Design Considerations

As stated earlier, the goal of this document is to examine the Flash platform with an emphasis on its potential as a MICS research tool. This research can occur during any stage of the design lifecycle, from requirements analysis and problem definition, through concept design and development, to experimentation and analysis. MICS research has traditionally focussed on information systems within the Maritime domain, encompassing tactical, operational, and strategic perspectives. Frequently this research necessitates unique visualizations and user interface prototypes operating within a tactical context and with varying levels of fidelity. To clarify, 'interface' in the context of this paper refers to human computer interface, or more specifically the graphical user interface (GUI) we use to interact with applications running on modern information appliances such as desktop computers, laptops, mobile phones, tablets, game consoles, and embedded systems.

The design of any computer interface will be heavily dictated by the hardware platform it is targeted for; this is especially true in light of what input and output peripherals are available. The interface for an iPhone with its multi-touch input and small portable display will offer a very different look and feel than a triple-display workstation complete with mouse and keyboard. The scroll-wheel on modern mice is not present on trackball peripherals (popular onboard moving platforms like ships) – how does this impact the navigation behaviour of an application's interface?

The hardware in this context refers to information appliances; essentially any computer devices which can be used to manage information and includes desktop, laptop, consoles, and handhelds. It is important to note the hardware includes all peripheral devices and sensors which can be used by people to interact with the information appliance. These hardware peripherals play a direct and defining role in the design of the GUIs which they communicate with. For example, a GUI which is driven in part by a mouse with a scroll wheel will have event driven behaviours specific to the scroll up and scroll down actions available from the mouse wheel. Likewise a multi-touch capable device like the iPhone enables new and unique gestures (e.g., pinching) which result in unique input events and subsequent behaviours of the interface. If we further consider device capabilities such as web cams, eye-tracking, accelerometers, voice recognition, motion tracking, multi-touch, and any other of a number of growing biometric style input devices, then we ourselves become input peripherals with unique input capabilities to control the communication with our digital devices (Figure 58).

The NeuroSky MindSet is an example of a newer category of computer input devices known as brain-computer interfaces. The MindSet is one of a number of emerging consumer level devices capable of sensing human brainwave activity. Recently an open source ActionScript 3 API for developing Flash applications that harness input from the MindSet was created by veteran Flex developer Sean Moore. Moore has also created a couple simple Flash based visualizations which are driven by data streamed from the Mindset. Links to the APIs, documentation, and a video demo are available on the SeanTheFlexGuy blog.

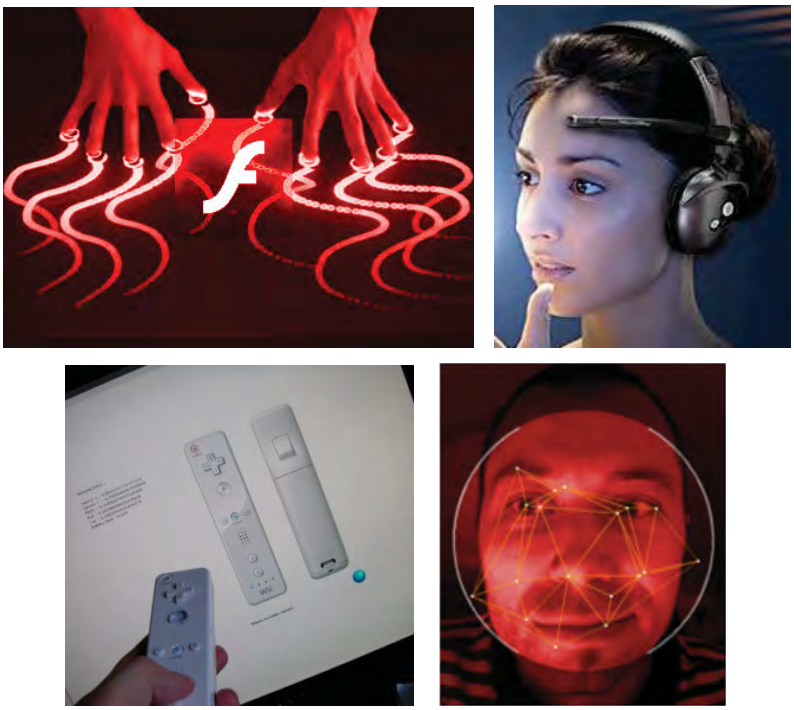

Figure 58: Native and third-party ActionScript APIs for emerging input devices like multi-touch displays, brain-computer interfaces, Nintendo's Wiimote, and computer vision systems further expand Flash's potential as an interface prototyping tool.

Nintendo's wireless Bluetooth remote for the Wii console known as 'Wiimote' is another unique and innovative input peripheral which was quickly adopted by Flash developers. The WiiFlash project and community is dedicated to developing Flash applications which can be controlled using the Wiimote. This includes maintaining the WiiFlash server (java/.net) for handling Wiimote communication and a WiiFlash ActionScript API for developing client applications.

The Machine Vision[92] demo on Flash developer Tomek Augustyn's blog was developed in Action Script 3. By analysing recorded or live video feeds, the web application can identify facial features and gestures in real-time. These types of computer vision systems can enable new and unique interaction capabilities between people and machines. A simple example might be a user monitoring system which automatically detects attention losses (e.g., operator fell asleep), mood changes, or perhaps an application which scales or positions display content based on the user's gaze or head position.

The performance characteristics of the host hardware, or more specifically the CPU and GPU processing power, will also heavily influence GUI design. For example the use of animations, graphical effects like transparency, drop shadows, texture resolutions, and 3D effects will be completely different between the GUIs of a desktop PC and a smartphone.

The output capabilities of information devices play a similarly important role in the design of an interface. The look and feel of an interface may vary dramatically depending on whether or not it was designed for a 3D display, a large wall display, or a small handheld display. Thus it is

important in the assessment of any interface prototyping tool to consider not only its ubiquity across the spectrum of existing and emerging information appliances, but also how well it supports each device's specific input and output capabilities.

From a general construction perspective, the user interface has both a 'look', and a 'feel'. The 'look' is largely a graphical design and layout problem, while the 'feel' is how it behaves – its interactivity. In a purist stereotypical world, the form is the realm of artistic designers, while the function is left to the more logic-oriented developers. It is interesting to note that form and function are in fact deeply related. Using the simplest of examples, imagine a piece of text on an interface – how would you know it functions as a button? Putting lucky guesses and previous experiences aside, the familiar and proven means for recognizing a button is largely related to its appearance; a rounded shape, a conspicuous location; subtle shadows, and raised surface. The bottom line is that form is intimately tied to function; if it doesn't look like a button it will not function well as a button. The reverse can be even more obvious, if for example an app is not capable of tool-tips then much more real estate will likely be necessary for labels.

Other factors which influence GUI design include:

- network connectivity – stand-alone, intranet, internet, or occasionally connected, and connection quality (e.g., bandwidth and latency);
- browser or desktop architecture – thin or thick client, deployment (e.g., installed or loaded?), local resource access (e.g., can user data persist between sessions?);
- security and privacy – connectivity and data format constraints, encryption and authentication requirements, field of view restrictions, and multi-user limitations
- ambient conditions – dark or bright, quiet or noisy;
- operational atmosphere – tension, criticality, and tempo; and
- user proficiency – age, demographics, experience level, previous training, usage frequency.

The last point regarding user experience and familiarity with common standards is a particularly critical factor influencing interface design. For example, Windows users expect a File menu to be accessible from the top left area of an interface. A floppy disk symbol universally denotes a save file function. This leveraging of pre-knowledge is even more important with regard to the web. Websites are front-end clients for direct human interaction; they are by design a form of user interface. In fact it could be argued that (conventional) web pages are now the most common GUI people deal with on a daily basis. Although there is great diversity among the look and feel of many websites, there is more often than not a great deal of standardization in terms of controls, layouts, menus, and navigation behaviour. Controls and behaviours that have become standard in the browser inevitably proliferate to other application interfaces. For example if a piece of text with underlining always acts as a clickable navigation link (e.g., a hyperlink), then users will come to expect this same behaviour in any application, regardless of the platform or context. Another example is the browser's back button, which users generally have specific expectations with regard to its look and function. Within a research context it is therefore important to consider established or emerging trends on the web when designing any new or experimental interface.

As eluded to earlier, this paper considers interface design largely from the context of military information systems; and more specifically those used in present and future naval platforms. However, it is still important to consider the complete spectrum of information domains including entertainment based interfaces like those used in gaming consoles. One reason for this is the transfer of technology often follows from these highly competitive markets to other markets including the military. For example, if interface technologies such as multi-touch and augmented reality prove themselves in consumer markets, it is quite likely such hardware will proliferate to the military domain. User experience also follows this same cross domain migration. The same kids now familiar with the Nintendo Wii remote will be tomorrow's UAV operators; guess which input interface they will be pre-trained and most familiar with?

3.4.2 Interface Prototyping

Research within the MICS section at DRDC Atlantic has historically included a diverse set of topics related to capturing, simulating, consuming, sharing, and analysing information critical to military operators and tactical decision makers. Many of these projects manifest as unique visualizations and user interfaces prototypes with varied levels of fidelity, often requiring prototyping tools capable of highly customizable interactivity and rich graphical animations.

Flash is largely a client side user-facing technology, and with its extensive set of GUI design and development tools (like the Flex framework), it is well positioned as a platform for human computer interface (HCI) design, development, and experimentation. As previously discussed, graphic design and animation is a core strength of Flash, and tooling such as Flash Professional offer a plethora of features specific to visual design and layout. Interface behaviour is built-in to many UI controls and can be easily customized, extended, or added as necessary using Flash's programming languages ActionScript and MXML. Suffice to say Flash tooling can be used to construct and/or modify experimental user interfaces in nearly any manner desired.

Adobe Flash was successfully used for prototyping a number of novel visualization and tactical interface concepts developed within the MICS Section at DRDC Atlantic (Figure 59). Flash based prototypes provided a more 'hands-on' means for researchers to explore and refine various proposed interface concepts. The ability to replicate HCI ideas into interactive models with high levels of custom interactivity, as well as the ability to quickly inject operational context through the use of custom animated scenarios was very beneficial to the overall design process. These same software artefacts can be further employed as an effective mechanism for sharing and soliciting feedback from subject matter experts (SMEs).

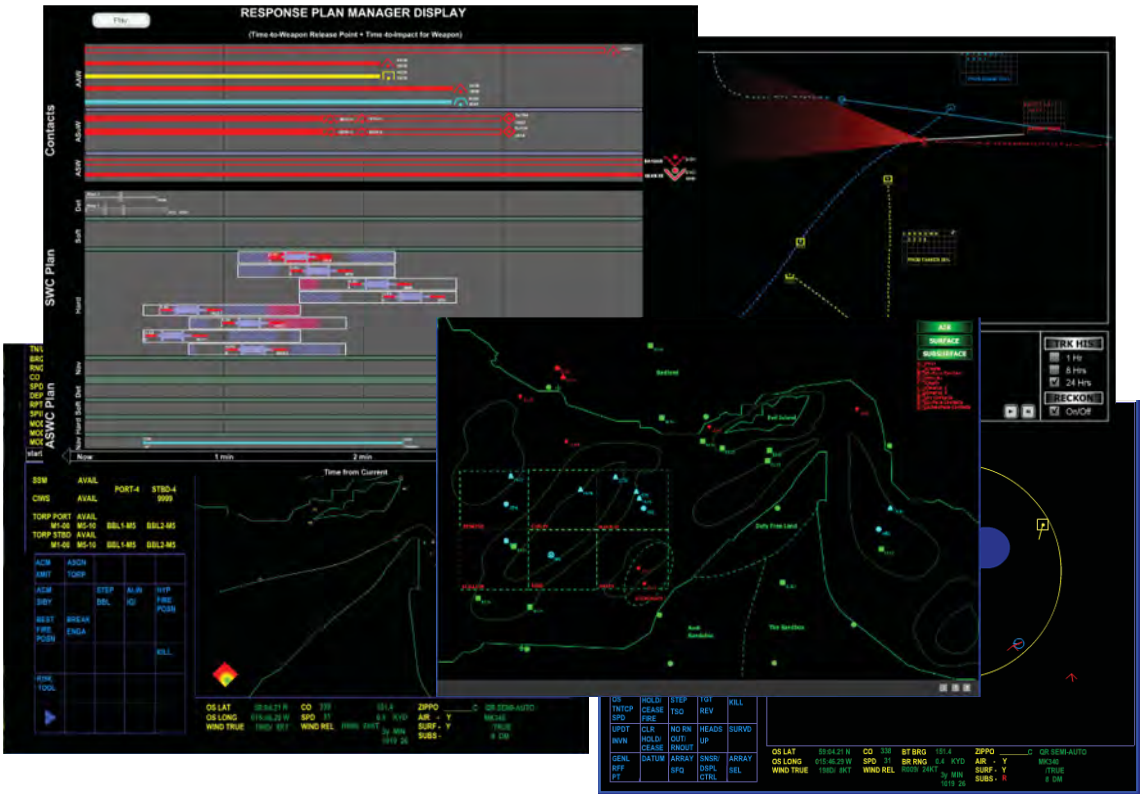

Figure 59: Flash based prototypes of novel tactical visualizations and interface concepts provide effective research tools for design exploration, refinement, and validation (screen captures from a selection of tactical interface demos used by Defence researchers at DRDC Atlantic).

3.4.3 Experimentation

Experimentation in a general sense refers to the testing of an idea or hypothesis, or more specifically within the context of this paper, using Flash as a tool for HCI testing. Interface design, whether revolutionary or evolutionary, is as much about look and feel as it is functionality. Functionality is largely a technological problem which is relatively easy to test (e.g., either you can update records in a database or you can not). On the other hand, the human side of interface design is an entirely different and much more difficult challenge.

Often the best means for collecting HCI metrics such as usability is to have real people using the software and observe, record, and analyse task performance as well as collecting user feedback. Thus a consistent method for collecting or logging user performance metrics is an important pre-requisite capability for experimentation tools. ActionScript is an event-driven language in which any and/or all events, whether user or system initiated, can be easily recorded at virtually any interval or frequency desired. Depending on data collection requirements, Flash developers can

choose between writing their own logging routines from scratch, utilizing built-in logging API's, or leveraging third party logging and/or analytic tools (Figure 60).

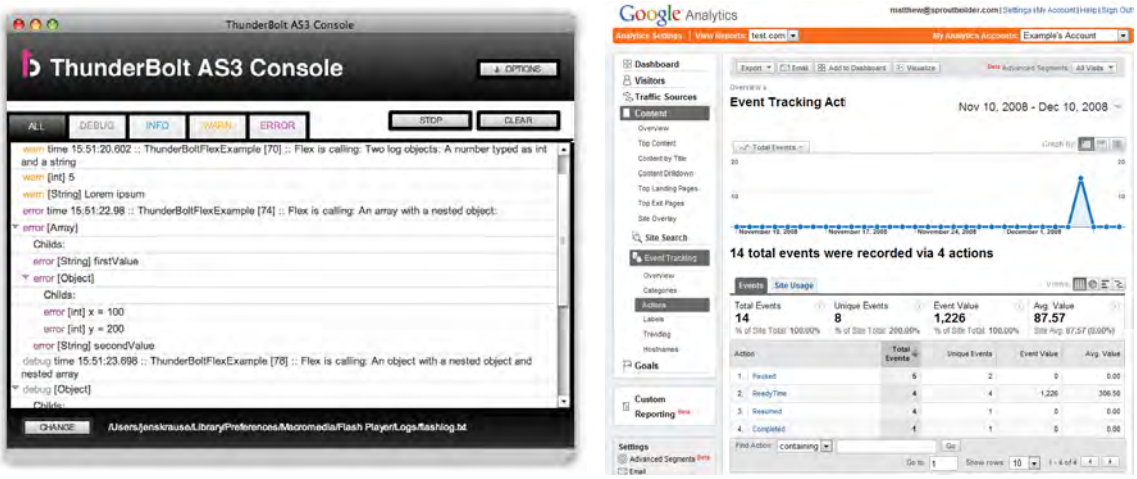

Figure 60: ThunderBolt AS3[93] is a free open source tool that enables extending logging capabilities within Flex, AIR, and Flash applications. Google Analytics software can be leveraged for tracking activities within Flash web applications using the gaforflash[94] free open source API.

The ubiquity of Flash applications across the growing spectrum of information appliances including everything from smartphones and embedded systems to web and desktop systems offers a truly diverse infrastructure for testing usability. Support for diverse input controls such as multi-touch, web cams, voice, and Wii style remotes are just a few examples of the breadth of choices available for user interface experimentation.

Being a web based technology, Flash can be used to architect web based experiments. For example, previous research conducted at the University of Warwick[95] has shown Adobe Flash can be used to run complex psychological experiments over the Web. In another recent experiment[96], virtual keyboards used for text input on mobile devices were constructed and evaluated using Flash. While classic experimentation typically involves the rigidly controlled environment of a physical laboratory, virtual online web based experiments can offer unique advantages. From a subject's perspective, accessibility is greatly improved by being able to participate at a time of their convenience and from the comfort of their own home. Experimenters avoid the significant logistics and resources required for onsite sessions including scheduling, chaperoning, facilities orientation, familiarization with lab equipment and policies, etc. Consequently, the number of subjects used can be orders of magnitude larger.

3.5 Mapping

Maps such as the navigation charts used by early explorers have existed long before computers, however their modern digital descendants offer many features and conveniences that are simply not possible with printed ink. Today's mapping applications can offer interactivity such as

panning and zooming, incredible resolution and detail, frequent updates, search ability, live overlays such as traffic and weather, multiple views including terrain and satellite, and even street level photo panoramas.

Flash is a popular client platform for building map based applications on the web, and Flash specific mapping APIs (and components) are available for all the major map service providers including Google Maps, MapQuest, Microsoft Bing Maps, OpenStreetMaps, ESRI ArcGIS, and Yahoo! Maps. These developer map tools greatly simplify and enhance the development of GIS based applications which incorporate the information and services offered by map providers. Adobe's Tour de Flex sampler application includes dozens of mapping examples from multiple map providers (Figure 61) and is a good source for a quick sampling of what's readily possible with regard to Flash based map applications. For example, drilling down into the MapQuest Maps and selecting the GeoRSS category will load the earthquake map shown in Figure 62. This particular example overlays near-time Earthquake information from an RSS feed and includes event time, epicenter location, and magnitude.

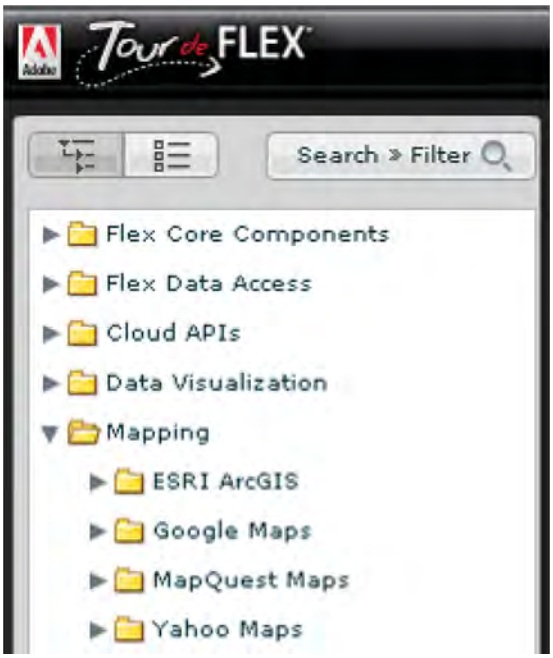

Figure 61: Adobe's Tour de Flex sampler provides numerous mapping demos from multiple map providers.

Figure 62: A Flash based MapQuest demo using a GeoRSS overlay – a near-time RSS information feed with embedded location attributes (e.g., earthquake data with epicenter and magnitude).

The VIPER application (Section 2.2.8.2) for situational awareness and emergency resource management is another excellent example of a map-centric application, in this case built using the ESRI Flex API. The Maps Room demo described in Section 3.2 is an example of a collaborative mapping application built using Google Maps, and an enhanced version with webcam support powered by Yahoo! Maps can be found within the collaboration showcase section of Adobe's website.

Finally developers may wish to look at the versatile UMap ActionScript 3 mapping API available from AFComponents. It offers integrated support for OpenStreetMap, Bing Maps, Yahoo! Maps, and ArcGIS map data providers. UMap is free for non-commercial use and the AFComponents website offers support forums, tutorials, and a customer showcase of applications built using UMap.

3.6 Physics

On the surface, physics may seem like an odd bedmate for serious application interfaces, but physics type effects have started to appear in newer and unique GUIs. A classic example is the throw slider used in multi-touch devices like the iPhone. Users can navigate large display lists using a finger swipe action which causes the list to scroll with both 'inertia' and 'friction' type behaviours. The power cursor demos (Section 2.2.7) are another great set of examples that use 'forces' applied to the mouse pointer to enhance the user's interaction various GUI controls. Newer multi-touch hardware is capable of detecting pressure and future interfaces could harness this sliding scale input to mimic real world force reactions. For example pressing hard versus soft may incur different 'physical' reactions within the user interface – items may displace farther or faster. Similarly, imagine a scrolling interface in which the scroll speed is proportional to touch pressure. The zooming and panning behaviors of map based interfaces might be influenced by simulated inertia and friction forces. Stephen Weber demonstrates a simple vertical scrollbar with throw physics[97] on his blog; the interaction is very natural and it is puzzling why such behavior has not become standard within all applications. Another example might include a multi-touch table top photo viewing application which allows users to 'flick' pictures with similar inertia and friction effects.

Advanced visualizations sometimes utilize physics effects like inertia, friction, and elastic forces to enhance interactivity. The Molecule Viewer from the Tour de Flex sampler is a linked node visualization which uses 'elastic' connections between nodes and 'inertial' movement of the nodes themselves to improve interactive viewing of complex structures (Figure 63).

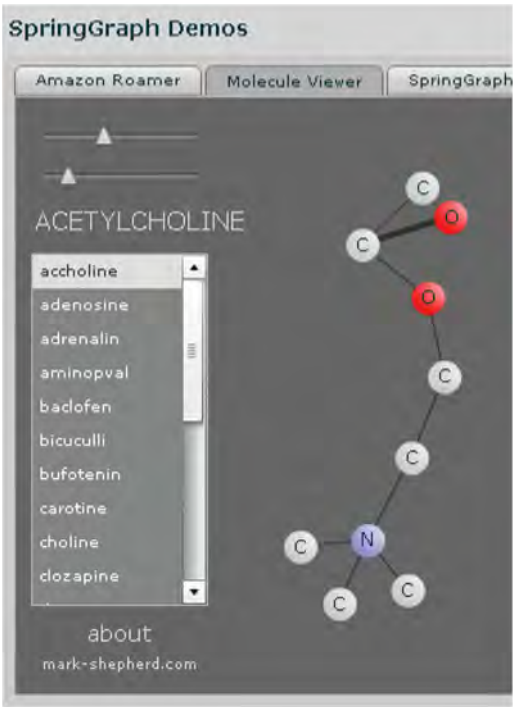

Figure 63: The Molecule Viewer is an interactive visualization which utilizes physics style forces.

Physics and 3D graphics are a very natural and popular combination because the 3D graphics typically represent real world objects and there is an expectation they will behave as real objects. The problem with 3D is its lack of proven functional value in web and desktop applications. The emerging segment of augmented reality applications may alter this premise. Because AR is by definition a view of the real world with graphics overlaid, the use of 3D graphics has potential because the real world is of course 3-dimensional. Therefore the 3D overlay graphics in future AR interfaces might incorporate some behaviors influenced by physics. For example, imagine a virtual search and rescue AR marker which 'floats' on the real world surface of the ocean.

Physics APIs allow developers to simulate the physical forces of nature such as gravity, friction, elasticity, inertia, and so on. For example an 'elasticity' class could be used to give an object such as a ball a bounce behavior. There are a handful of physics 'engines' for Flash which can be loosely categorized as either 2D or 3D. Understandably, physics APIs are most often associated with 2D and 3D drawing APIs as it is these virtual objects which often need to simulate real world behaviors. A few of the more popular physics engines available for Flash are listed in Table 5.

Table 5: Flash 2D and 3D Physics Engines.

Name	Availability	Notes
JiglibFlash [98](3D, AS3)	Open Source	Supports Away3D, Sandy3D, FIVe3D, and Papervision3D; very active development status.
WOW-engine[99] (3D, AS3)	Open Source	Supports Away3D, Sandy3D, and Papervision3D; uses APE 2D; no recent updates.
The Fisix Engine [100](2D)	Free[2]	Supports particles, rigid bodies, constraints; targeted at game developers.
Polygonal (haXe, 2D, AS3)	Open source	Written in HaXe, includes physics and computational geometry.
APE (2D,AS3)	Open Source	Very little documentation, no recent updates. Supports particles, collision, surface friction.
Box2DFlash (2D,AS3)	Open Source	AS3 port of C++ 2D physics library.

A good round-up of Flash 2D and 3D physics engines complete with brief descriptions is also available on Emanuele Feronato's blog posting 'Flash physics engines galore'. Also note live demos which utilize Flash physics engines are included in Sections A.1.9 ('3D Physics - Dice') and A.1.17 ('Interactive 2D Physics') of Annex A.

[2] "We are currently offering the Fisix Engine free for use for both commercial and non-commercial applications until our next release."[98]

3.7 Visualization

Visualization is basically the process of using graphics to represent data with the goal of making the data easier to understand. The human visual system is an incredibly fast and powerful processing component of the human brain; vision represents the bulk of sensory bandwidth connection to the brain and significant portions of our grey matter are dedicated to vision. Data visualization essentially aims to leverage the power of the human visual system to enhance our ability to make sense of the data; recognizing trends, anomalies, and other visual patterns can become orders of magnitude faster and more effective when viewing data depicted in graphical form.

Depending on the Flash development tool used, often some basic visualization components (e.g., simple charts) will be included by default. For example, FlexBuilder 3 includes several standard visualization components such as bar, area, bubble, pie, and other common chart types. However larger and more complex data sets may require more advanced visualizations including dynamic and interactive capabilities.

For those instances where more sophisticated visualizations are desired, there is an abundance of third-party visualization tools, components, and APIs available freely and by purchase from several online sources (see Section 8.6). In addition to the standard out-of-box charts, Adobe's Tour-de-Flex sampler application includes dozens of advanced visualization examples created using Flex APIs such as Axiis, IBM ILOG Elixir, Kap IT Lab, and Mindset Designs (Figure 64).

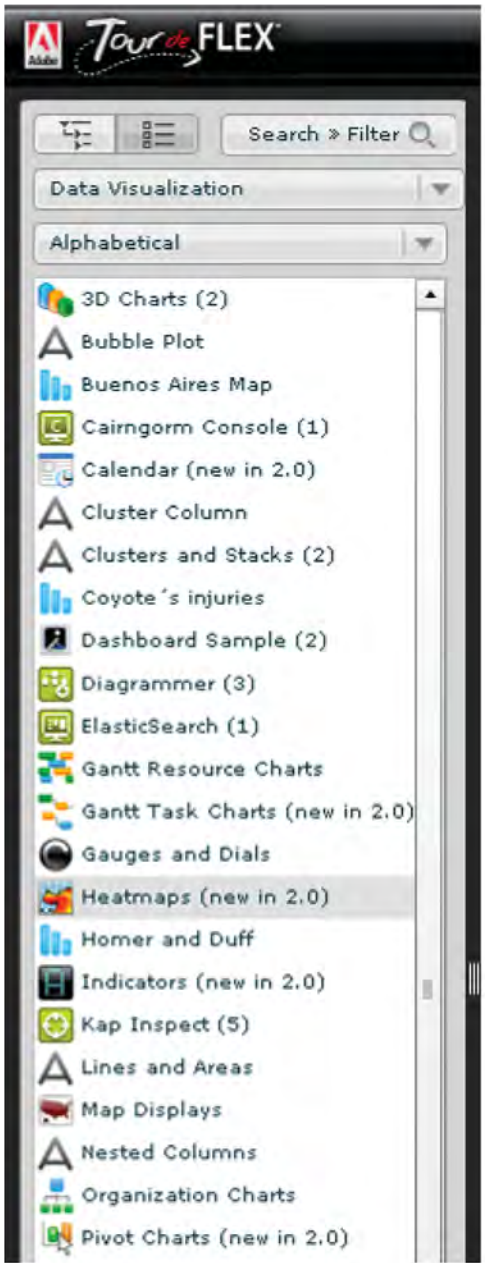

Figure 64: Adobe's Tour de Flex demo provides many advanced visualization samples.

Flash is a natural fit for creating and viewing data visualizations, as graphics, animation, and interactivity are core strengths of the Flash player. This fact is reinforced by the shear number and variety of Flash based visualizations (Figures 64-69) and third-party tools in existence. Data visualization is also a core recurring research interest within the MICS section at DRDC Atlantic. The general requirement for tools to aid in making sense of data in ever increasing amounts and complexity within the Maritime theatre remains a strong research opportunity. The exponential growth of sensor capabilities, data harvesting systems, and

information systems in general increasingly highlight the human in the loop as the weak point. The value of properly designed visualizations can not be over emphasized. Modern technology in many forms including scientific equipment, information appliances, complex sensor networks, and other data harvesting systems is creating information faster than our ability to comprehend it. Simply put our increasing ability to produce and collect data continues to outpace our ability to understand it. Information tools that can reduce the cognizant effort required by operators and decision makers are universally welcome; this is especially true within the battlespace.

It is no secret that visualization is very much an art as much as it is science. The fact the Flash platform has strong tool sets for both designers and developers facilitates the creation of effective graphical representations of increasingly large and complex data sets.

Visualization has been around for a very long time within traditional print media, and many early computer based visualizations simply replicated this capability. However, computers also bring new capabilities to visualizations which are simply not possible with ink. These advanced computer visualizations leverage computer specific features such as animation, 3D, and interactivity. It is these types of advanced computer visualizations that have the most potential as visual aids for understanding increasingly complex and disparate information streams.

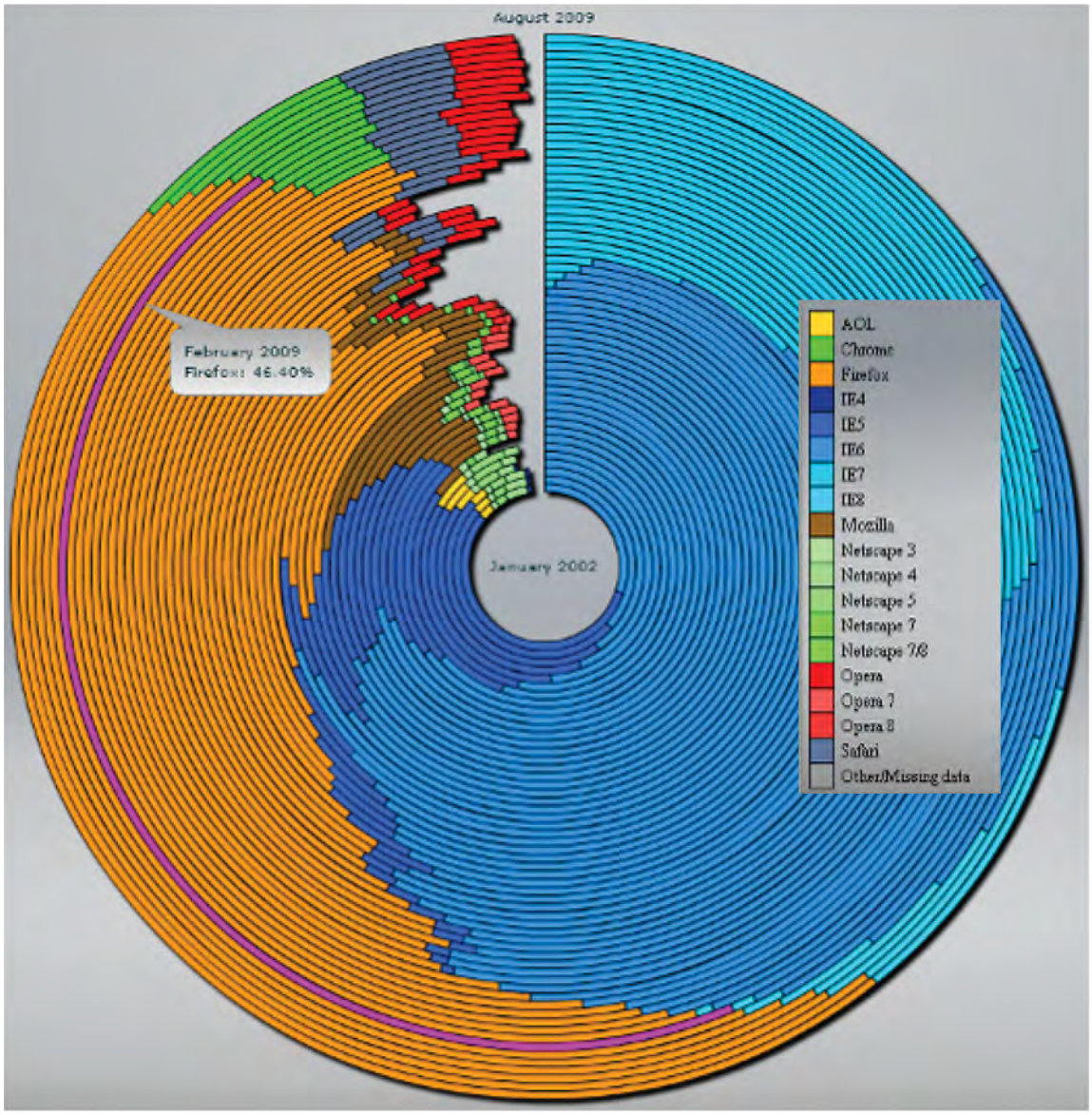

Figure 65: A growth-ring visualization, created using the Axiis[101] API, shows the relative popularity of competing web browsers over time. The chart is predictably dominated by Microsoft's Internet Explorer and its successive versions (blue segments), as well as Mozilla's Firefox (orange segments). The recent introduction of Google's Chrome browser shows up as the fluorescent green on the outer "11 o'clock" ring segments.

Figure 66: Visualization examples created using the open source Axiis[101] framework for Flex.

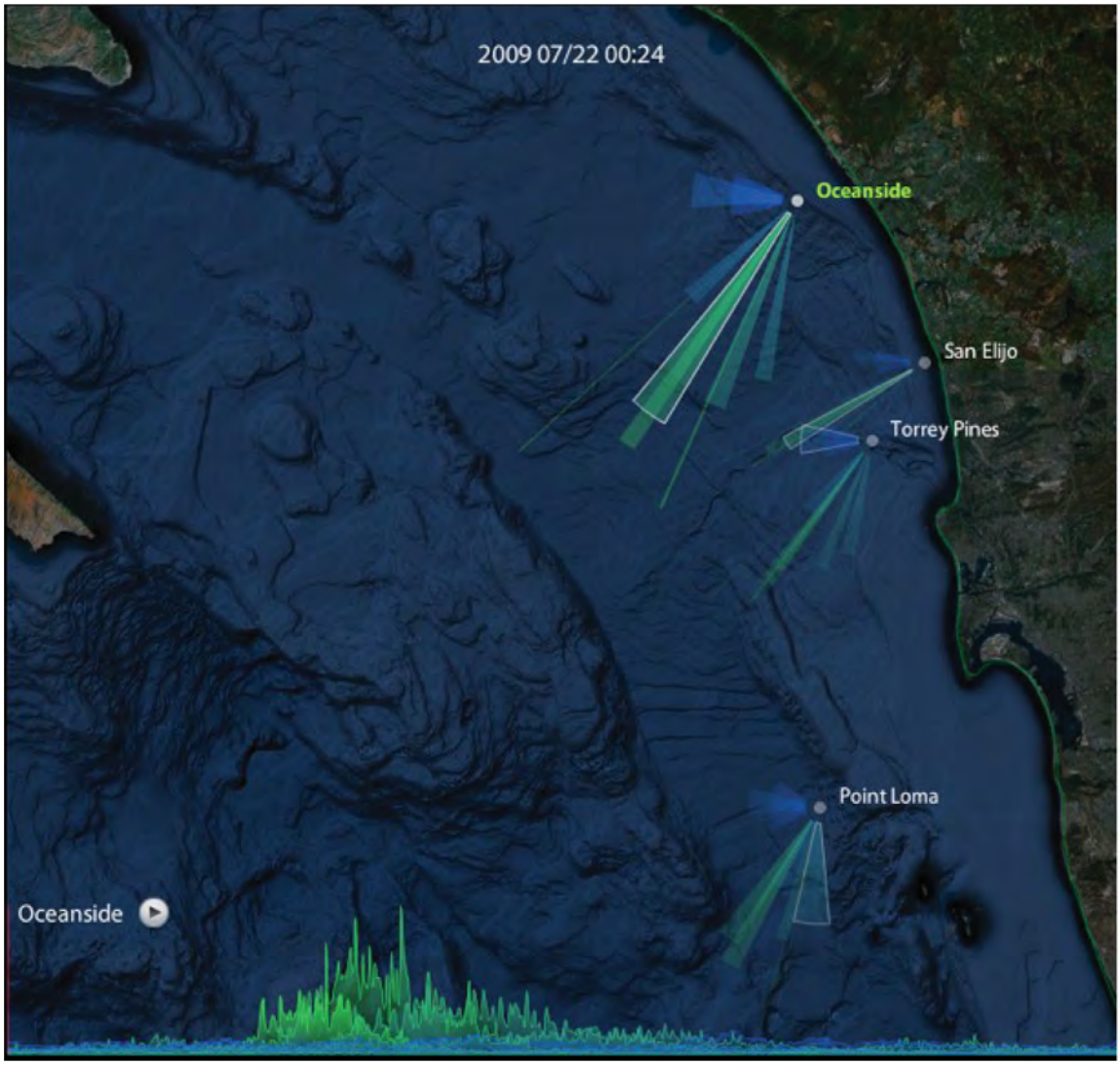

Figure 67: The wave data visualization[102] represents frequency, direction, and strength of wind waves (blue) and ground swell waves (green) over time. A histogram of wave strength provides temporal navigation in addition to longer term trend patterns. In this case a significant ground swell event is easily discernable along the center segment of the timeline. The actual Flash application is included in Section A.1.18 of Annex A.

Figure 68: A 'heat' map (with histogram) created using SpatialKey[103] commercial software shows annual graffiti counts in NYC.

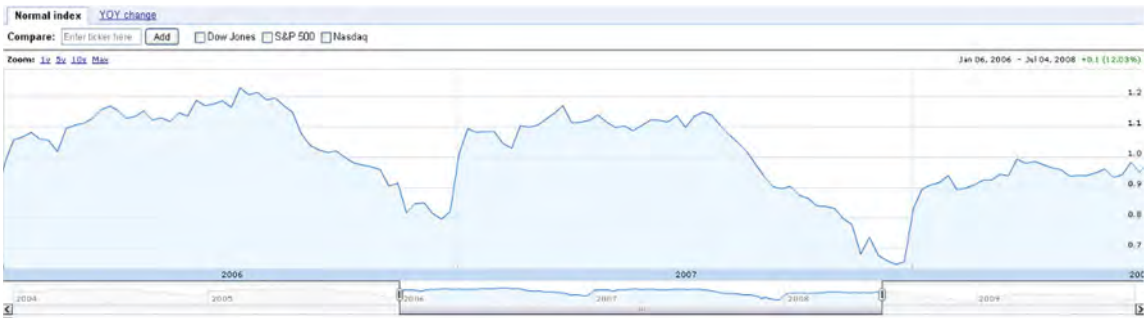

Figure 69: Google Finance[104] uses Flash to create advanced interactive visualizations of trends in the finance markets. Note how the thumb slider is dual purpose – pan/zoom navigation and histogram.

Figure 70: An interactive spatial-temporal visualization[105] of daily mass transit usage patterns on the MBTA (Massachusetts Bay Transportation Authority).

For further examples and resources related to data visualization using Flash, readers can refer to Section 8 'Resources Links'. Additionally, Section A.1.14 of Annex A contains working Flash based visualization examples created using the Flare[106] toolkit which readers can interact with.

3.8 3D Applications

Typically 3D applications are characterized by vector graphic objects with three virtual dimensions (e.g., x, y, and z) rendered on a physically flat 2D display. At first glance exactly what qualifies as a '3D' application may seem somewhat obvious. However consider an application like Google Street View (Section 2.2.2) in which users navigate a virtual, semi-3D space 'painted' with flat photos and the division between 2D and 3D is less distinguishable. Many augmented reality applications overlay 2D graphic icons to mark locations in real world 3D space when viewed through a flat digital display. Other applications allow users to position or manipulate 2D images or surfaces in 3D space. Google Maps 3D is one such example which enables users to control the pitch, yaw, and roll of a 2D map in 3D space. Another similar example is the Hotels and Events website; a web application for booking hotels which combines a 3D view of a 2D map and adds 3D location markers.

Image carousels are another example of positioning 2D objects in 3D space (Figure 71). In fact with the release of Flash Professional CS4, Adobe introduced partial native 3D support in the form of 3D translation and rotation tools which can be used to manipulate 2D surfaces in 3D space, also known as 'postcards in space'.

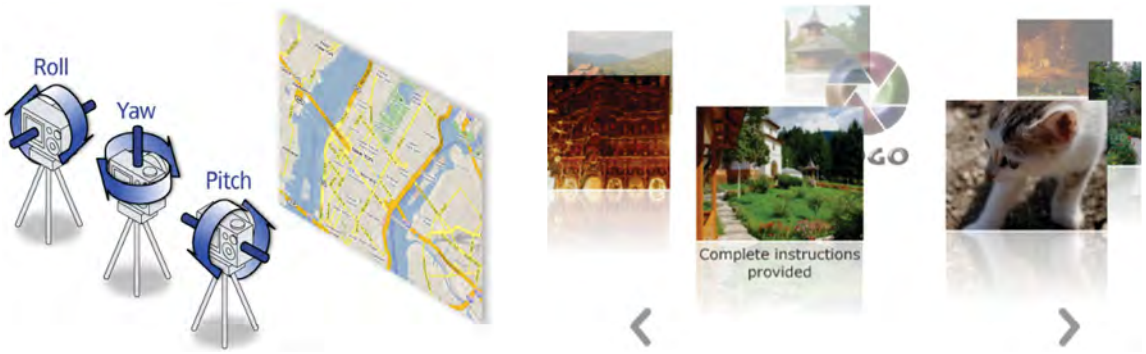

Figure 71: The Google Maps API for Flash[107] supports 3D viewing, enabling viewers to manipulate 2D maps in 3D space (left). A carousel enables navigation of 2D images in 3D space (right).

None of these examples can be considered true 3D applications in the traditional sense. In the case of fully 3D graphic environments, it is important to first note these types of applications are particularly demanding in terms of hardware requirements. Hardware acceleration is a method of offloading portions of the CPU processing load to the GPU and ultimately improve the rendering speed and responsiveness of graphically demanding applications. Generally speaking Flash performance substantially lags in comparison with native desktop 3D applications. A large contributing factor to the limited 3D rendering speed of Flash applications is no doubt tied to the lack of 3D GPU acceleration. If you are familiar with the graphic rendering quality and performance of previous generation gaming consoles such as the Playstation 2 or Nintendo 64, then you have a good sense of the current state of 3D Flash based graphics.

Adobe has been incrementally introducing specific and limited hardware acceleration to the Flash platform. As of Flash player 10, Adobe included GPU hardware acceleration claiming to "accelerate compositing calculations of bitmaps, filters, blend modes, and video overlays faster than would be performed in software"[108]. Uro Tinic, one of the Flash player's engineers, further clarifies exactly what Flash player 10 hardware acceleration means in a 2008 post to his blog site[109]. In short the added GPU acceleration functions are associated mainly with 2D graphics and video processing, and hardware acceleration of 3D graphics is still quite limited. Thus until the Flash player incorporates full hardware acceleration specifically for 3D graphics, expect 3D Flash applications to remain very CPU intensive, despite the presence of dedicated 3D hardware.

Despite the lack of relative performance, creating fully 3D Flash applications remains a popular endeavour within segments of the Flash development community. A number of third party Flash 3D tools, components, and APIs (3D engines) have sprung up over the years, partly out of necessity given the lack of native 3D support in Adobe's Flash design and development tools. The motivation behind creating and developing these 3D engines seems to be largely driven by the online gaming, entertainment, and advertising markets.

There are currently over half a dozen 3D engines for Flash, each being frequently updated with new features and support for newer Flash player versions. Thus information regarding capabilities, performance, and other features quickly becomes out dated, however a regularly

updated list of 3D engines is maintained on the “Flash 3D” Wikipedia article[110]. Additionally, Ding ZhiGang’s blog (ntt.cc) provides a round-up of several popular Flash 3D engines such as Papervsion3D, FIVe3D, Sandy 3D, Away3D, and Alternativa3D (Figure 72).

Figure 72: Tanki is a massive multi-player online (MMO) game developed using the Alternativa3D Flash 3D engine.

Most of these 3D APIs are open source licensing and can be downloaded freely from their respective websites, typically in the form of a ‘.swc’ (Section 5.1.1) library file. As always developers will have to add the downloaded swc file to the build (library) path of their coding IDE (see Section 5.1.1). These APIs more often than not are supported by an active community of developers who share FAQs, tutorials, links, sample code, and Q&A information via a community blogs, forums, and wikis. There may be multiple versions of the same 3D API with each being specific to a version of the Flash player (e.g., FP9 or FP10) as well as versions built using AS2 or AS3. Recall the Flash player includes 2 virtual machines, AVM1 for AS1/AS2 and AVM2 for AS3 code.

As mentioned, many Flash 3D examples are purely for gaming, however there exists a subset of this application domain with more practical implications, or more specifically 3D simulators and trainers. A perfect example of this is "Start Thinking Soldier"[111]; a hybrid 3D FPS and interactive video application that places you in the role of a commanding officer making decisions in the line of combat. It becomes difficult to distinguish such an application between game, training simulation, and recruiting website. Given none of these things are necessarily mutually exclusive; it probably satisfies criteria from all three categories in varying degrees.

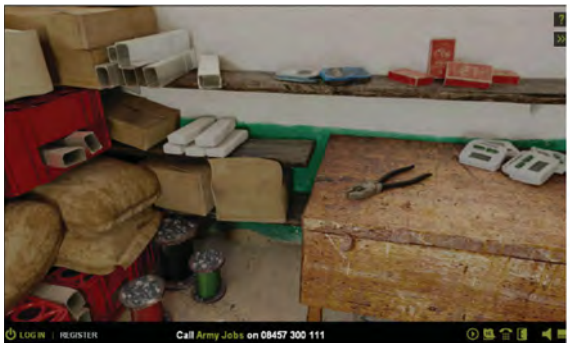

Figure 73: "Start Thinking Soldier" is a Flash based, hybrid 3D FPS and interactive video web application available on the UK Army's website[111].

Yet another area where full 3D engines have been utilized is in general website design and navigation. A collection of 3D examples used in web site design are listed at the DesignReviver website[112]. The "ecodazoo" website (Figure 74) is one such example in which full 3D Flash is used to construct an entire website. The site features an explorable 3D world for children, in a pop-up book style. It is based on the high performing Sharikura 3D engine for Flash, and features inertia and cloth physics effects to enhance interactivity.

Figure 74: The "ecodazoo" web site uses the Sharikura 3D Flash engine and features 3D graphical navigation with physics enhanced interaction [113].

Using National Geographic's Globe of Human History[114], site visitors can interactively navigate a 3D earth globe and learn about the known history of human migration. The globe was built using PaperVision3D and can be rotated in any direction by clicking and dragging with the mouse. The date range selectors on the timeline above the globe can be used to dynamically adjust the window of time applied to the globe.

Figure 75: National Geographic's Globe of Human History[114] uses an interactive 3D earth globe for navigation and visualization of civilization's history.

A final noteworthy application area for Flash and 3D is with regard to text. Version 3.4.1 of the Away3D Flash 3D engine enables developers to take advantage of the 3D text fields introduced with Flash player 10. Text can be aligned and animated along any 3D curve (Bezier), surface, or object (Figure 76). The example on the left consists of a 3D Bezier curve with handle points (red circles) which can be freely positioned in 3D space. The text dynamically flows along the line while maintaining 3D orientation, much like a roller coaster.

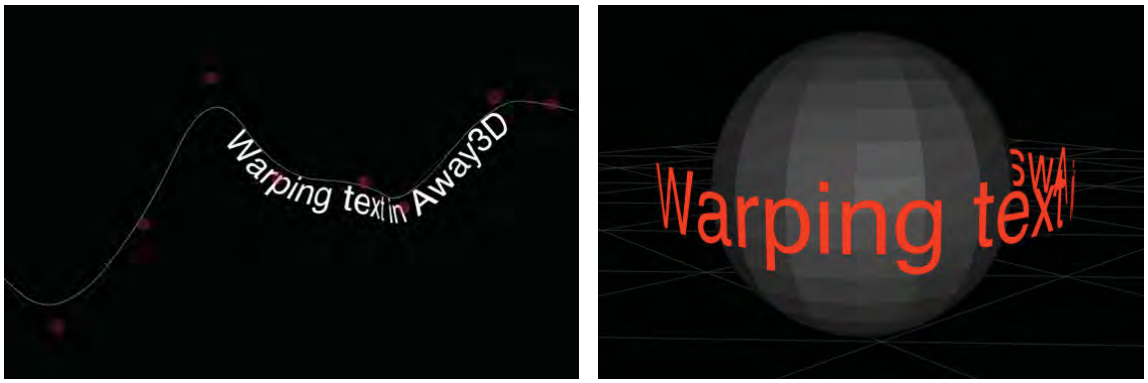

Figure 76: Examples of 3D vector text fields created using Away3D[115]. Animated text flows along a Bezier curve in 3D space (left). Animated text flows around a 3D sphere (right).

3.9 Augmented Reality

Augmented Reality (AR) refers to the process of enhancing a person's view of the real world by superimposing it with computer generated graphics and information. This augmenting information which is digitally 'injected' into the person's view provides additional contextual awareness about the objects currently being viewed. It is nothing new if you consider examples such as fighter pilot HUD displays, targeting-assistance systems, or the yellow first down line displayed during televised football games. AR applications will vary based on the hardware platform (e.g., handheld, desktop, TV broadcasts, game consoles, and embedded systems), the type of digital data overlay (e.g., 2D or 3D graphics, video, text), and the input controls (e.g., biometrics, motion tracking, marker tracking, GPS, accelerometer, webcam). While expensive and highly specialized professional/industrial grade AR systems (Figure 77) still exist within niche markets, AR technology is currently enjoying a rapid emergence into mainstream consumer platforms such as gaming consoles, desktop computers, and smartphones.

Figure 77: A military mechanic wearing a tracked head-worn display performs a maintenance task on a Rolls Royce DART 510 Engine (Left). A view through the head-worn display depicts information provided using augmented reality to assist the mechanic[116] (Right).

Currently there is plenty of ongoing research into augmented reality, and many entertaining and practical applications are beginning to emerge. This trend is further fuelled by advances in display technology, wireless communication, and general computing performance. Many of the current generation of augmented reality desktop applications fall within the amusing, entertaining, and/or just plain gimmicky categories with little to no functional value. While AR struggles for practical use cases on the desktop, 'location aware' AR applications (Figure 78) being introduced on many advanced smartphones promise more value added scenarios, particularly with regard to situation and location awareness.

AR applications on mobile devices can be location aware using built-in features like a GPS, compass, and accelerometer. Combined with a built-in video camera, these positional inputs are used by AR mobile applications to enhance the end user's situational awareness, akin to an automotive GPS. Furthermore, these devices can often tap into the internet and pull down even more relevant information specific to where the user is currently located and/or what they are

currently looking at. For example, this could be information about nearby facilities such as hospitals, ATMs, transit stations, or coffee shops.

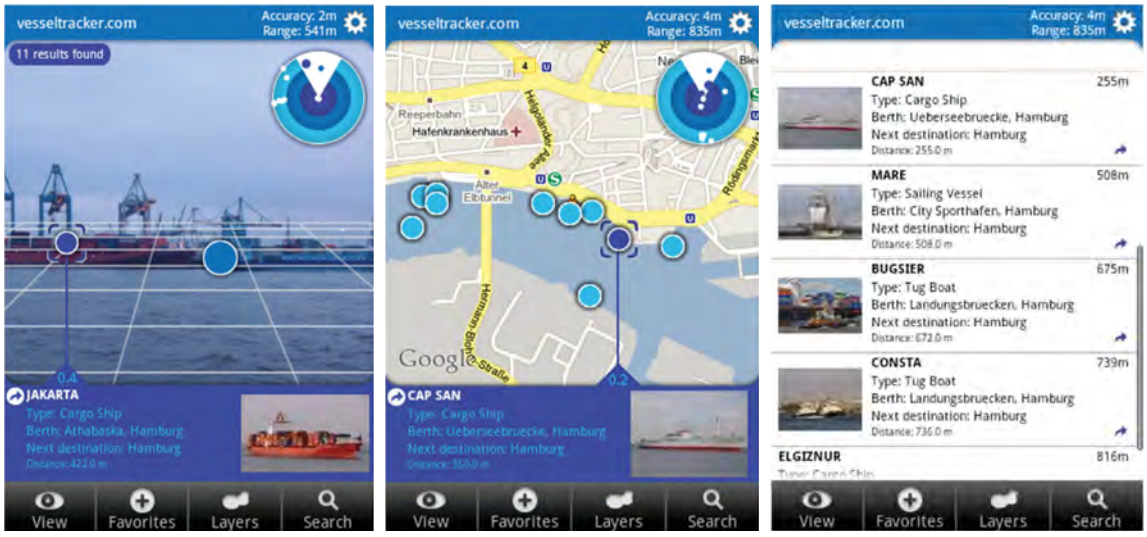

Figure 78: The Vesseltracker smartphone AR app displays "real time ship traffic and the positions of moored vessels in the world's largest ports powered by the vesseltracker.com AIS network"[117].

Although many existing AR examples are directed toward mainstream consumers, it is not difficult to imagine specific applications for frontline personnel (e.g., safety, security, and defence). Such field type scenarios could utilize virtual location markers (breadcrumbs) for search and rescue, surveillance, social networking, policing, and military operations. In other words, the AR system provides a kind of virtual annotation or tagging system of the real world, whether it be land, sea, or air.

It is worth noting AR and related technologies apply equally well to ANY apparatus used to view the real world. For example, AR capabilities embedded into viewing devices like binoculars, telescopes, microscopes, and periscopes could significantly enhance the observers' experience. Other applications could target the windscreen or navigation mirrors on various types of vehicles, whether they be land, water, or air based.

Flash based tools are often used for developing and experimenting with AR applications on desktop systems, particularly with the DIY crowd. This trend is probably due to the relative ease and low cost with which Flash based AR applications can be created. Many of the current Flash based AR applications use a marker based input control combined with a web cam like that shown in the GE Smart Grid demo[118] (Figure 79). Home users can interact with an animated digital hologram aimed at promoting GE's eco-friendly Smart Grid technology campaign. The GE demo additionally uses microphone input to control animations; for example the sound of blowing air across the microphone causes the windmills to spin.

Figure 79: The GE Smart Grid AR demo[118] is used as a fun interactive promotional tool (left). The Virtual Box Simulator[119] on the US Postal Service website helps customers determine shipping package sizes (right).

As a more practical example, the United States Postal Service's Virtual Box Simulator is an AR application which allows customers to accurately estimate the size of required shipping packages. Using a transparent virtual box, would be shippers can determine in advance exactly what size they will need for their specific contents.

FLARToolKit[120] is currently the preferred SDK for creating AR applications using Flash. Developing Flash based AR applications is relatively simple using either Flash Builder or Flash CS4. Lee Brimelow provides an excellent AR video tutorial[121] using Flash Builder on his gotoandlearn website. For Flash CS4 Professional users, the Adobe developer site has a great tutorial for creating an AR application.[122] In both examples you will need to download and import the FLarToolKit code library (to handle marker detection and tracking) as well as the Papervision3D[123] code library (to create and render the 3D graphics overlay). The FLARToolKit-userz group has an excellent portal website[124] with links to all things related to creating AR applications using Flash including tutorials, tools, and demos.

While it is relatively easy to use Flash for 'quick and dirty' AR experimentation, application performance can be relatively poor compared with natively compiled AR programs. Thus the latter may be more preferable when developing more advanced AR programs that involve more complex 3D visual overlays, higher webcam resolutions, and faster capture rates. This is even truer on current smartphone devices where hardware resources such as CPU/GPU power and memory are even more limited. This is indeed unfortunate as location aware AR applications for mobile systems seem to offer the greatest potential, particularly in the areas of real-time situational awareness. That said, there are a couple factors offsetting the general performance handicap of virtual machine platforms such as Java and Flash. The first relates to the rapidly increasing hardware capabilities of handheld devices in general; tomorrow's portable devices often equate to today's workstations in terms of computing power. The second performance factor to consider is brought to light by Flash's recent support for **native** iPhone applications. Using Flash Professional CS5, ActionScript based applications can be compiled as native iPhone apps with no requirement for the Flash player and its virtual machine. In theory, this 'write-once,

compile-anywhere' (WOCA) workflow and execution model could offer improved performance of Flash developed applications on handheld devices, however this remains to be seen.

3.10 Multi-touch

Touch driven interfaces are a class of GUIs which can be controlled by touching the display itself or an input surface. While single-touch input devices like single-touch pads are nothing new, affordable multi-touch systems in which more than one finger at a time can be used as input are a relatively new form of human computer interaction. With Multi-touch computer users are no longer restricted to just a keyboard and mouse, or single finger touch pads.

The launch of Apple's iPhone in 2007 signalled the introduction of multi-touch interfaces into the mainstream consumer market. Since then multi-touch capability has been steadily proliferating into desktop, tabletop, mobile, and handheld systems.

Multi-touch interfaces enable new and unique gestures such as tap, pinch, spread, swipe, nudge, flick, and rotate to name a few. More advanced touch systems are sensitive to finger pressure, enabling a sliding input scale from soft to hard. All these touch sensitive inputs combined with up to 10 fingers per person provide a plethora of new interface gesture possibilities (e.g., 3-finger swipe, 2-finger tap, soft pinch).

Ideum[125] is a company who designs and builds multi-touch displays and exhibits such as the interactive tabletop recently installed at the Ontario Museum of Science. Ideum also offer a Flash multi-touch framework and SDK commercial product known as GestureWorks[126]. The company claims GestureWorks-built applications "out perform applications using TouchLib by a 2 to 1 margin"[126] and its development tooling can greatly ease creation of Flash multi-touch applications.

Figure 80: A multi-touch, multi-user application allows visitors to explore Flickr photos placed on a Yahoo! Map at the Ontario Science Centre[125].

The arguments for multi-touch interfaces seem largely tied to their hardware platform. Those platforms that do not lend themselves well to conventional keyboard and mouse controls will likely benefit the most from a multi-touch interface. For example small handheld devices like the iPhone have such limited real estate for input, a double duty display and multi-touch input surface maximize the available input area.

Tabletop, wall, and kiosk displays are other examples of hardware segments where multi-touch makes sense, again given the unfeasibility of conventional keyboard and mouse input controls. It is important to note these types of interfaces are often used in multi-user collaborative applications. Multi-touch gestures are particularly well suited for manipulating images and maps. For these reasons multi-user interfaces like wall displays and tabletops may offer the most potential in research areas such as shared map and photo viewing, command and control, shared situation awareness, and own-ship collaboration.

As for personal interfaces such as traditional desktops and operator consoles where keyboards, mice, and trackballs are still feasible, the practical benefits of multi-touch based gestures seem unproven, despite popular Hollywood movie depictions to the contrary.

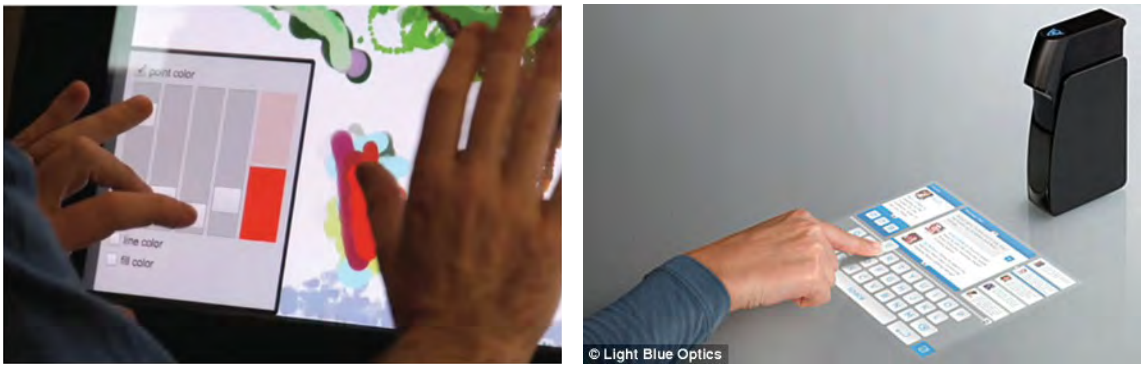

Figure 81: Adobe R&D staff *experiment with a multi-touch desktop application built using Flash[127] (left). The "Light Touch" by Light Blue Optics is a handheld projector and Flash based media player that turns any surface into a touch display (right).*

Due to the speed and ease with which client interfaces can be prototyped, Flash is a very popular choice for developing and experimenting with multi-touch desktop applications. Adobe has only recently introduced built-in support for multi-touch as of Flash player 10.1 and AIR 2.0. The 'Multi-touch and gesture support on the Flash Platform' article on the developer section of Adobe's website provides a good overview of Flash's native multi-touch capabilities. To take advantage of the new features, users must have both hardware and an OS which support multi-touch (e.g., Mac OS 10 or Windows 7).

For developing and experimenting with an OS or hardware not supported natively, the ideo-multitouch package for Flash[128] hosted on Google code contains a set of open source development tools including easy-to-follow instructions for building multi-touch applications. If you do not happen to have multi-touch capable hardware the ideo-Multitouch website also offers a handy Flash multi-touch simulator:

Touch events can be simulated in the Flash environment to allow multi-touch software to be developed without or while away from multi-touch hardware. Touch events are simulated by way of keyboard chording. The number keys place virtual fingerprints on the stage that can be dragged with the mouse. Visible fingerprints are also useful for debugging multi-touch hardware calibration and screen registration[128].

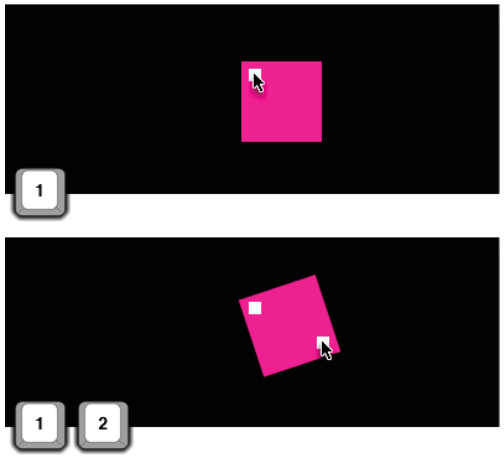

Figure 82: A Flash based multi-touch simulator enables multi-touch software development without multi-touch hardware[128].

Also worthy of note is the Natural User Interface Group (NUI Group); a community of developers dedicated to developing and experimenting with multi-touch applications. The NUI website is a hub of information related to multi-touch experimentation including discussion forums, video demos, downloads, and several do-it-yourself guides. The site's wiki includes an easy to follow tutorial titled "Building Your First Multi-Touch Application in Flash"[129].

3.11 Cloud Services

Cloud computing generally refers to the use of resources and services located on the internet (the cloud) to build and deploy software. In recent years there has been a growing migration of IT infrastructure, services, and even entire applications from local systems and networks to more centralized and shared providers, typically internet based. The trend is largely driven by the reduced costs associated with only buying as much IT resources and services as and when needed, in contrast to the upfront and overhead costs associated with in-house solutions. Scalability is another common argument in favour of cloud based computing; additional storage, networking, and computing resources can be quickly and easily purchased as required. Not having to worry about additional server management headaches relating to network configuration, security updates, disaster recovery, load balancing, and failover further enforces the appeal.

For developers, accessing and leveraging cloud based services via standard APIs can greatly simplify the development process. Consider the example of developing an application that includes a mapping component. Instead of building, deploying, and maintaining a proper map server, a developer can quickly and easily add mapping capabilities using a map service provider's API. More importantly however, there are a growing number of services and resources which are exclusive to the web including massive repositories of community-driven content on websites such as youtube, flickr, twitter, Wikipedia, and facebook. Other powerful web services include search, mapping, weather, traffic, news, email, and messaging to name but a few. Providers expose these services to developers using standard libraries (APIs) and tools targeted for specific web software platforms like HTML, AJAX, and Flex. From the client side, Flash development tools such as Flex Builder have adopted many developer features which greatly simplify the integration of cloud based services.

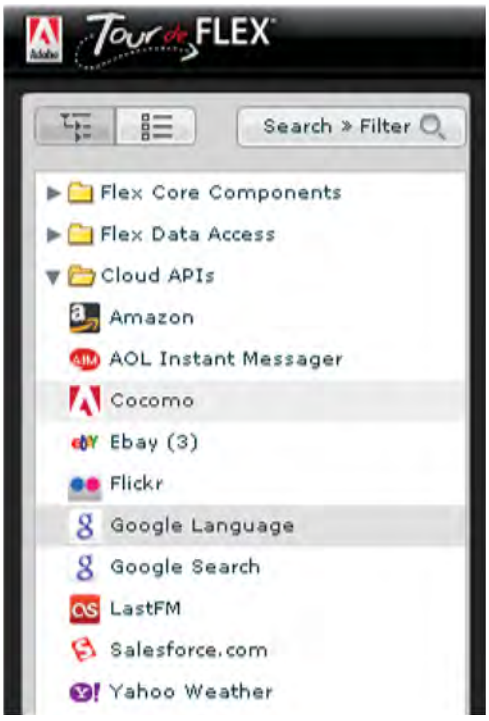

Figure 83: Cloud based service providers use custom API libraries to expose their services to developers.

As with most things available from the web

there is both an extreme abundance and diversity of offerings, many of which would have little to no value in terms of software development and experimentation within a Defence research context. Having said that, there are several conceivable instances where using cloud APIs could be beneficial. Such cloud based services could be leveraged to save time and complexity in research and experimentation areas such as situational awareness, collaboration, and interface design. Examples include Flash specific APIs for mapping (Google, Yahoo, MapQuest, ESRI), social networking (FaceBook, MySpace, twitter), collaboration (AIM, Google Wave), search (Google, Yahoo) and weather (Yahoo! Weather, Google Weather). Imagine a maritime tactical display concept which incorporates a real-time weather overlay component. Developing, integrating, and experimenting with such a component is significantly simpler and quicker using a cloud API then trying to otherwise replicate a meteorology service.

Figure 84: Many of the situational awareness feeds in the VIPER emergency management Flex application (Section 2.2.8.2) are exclusive web services. VIPER represents a class of mash-up applications which could not exist without the internet.

4 Developer Tools

4.1 General

Today, more than ever, there is an abundance of software tools available for creating Flash applications. While Adobe maintains the lion share of recognized Flash tooling, there are numerous alternatives available from both commercial and open-source providers. In all cases the selection, quality, and extent of features continue to grow and mature at a rapid pace. It is not the intent of this section (nor is it practical) to review all these tools in depth, nor provide a how-to training tutorial for each one. However, some of the more recognizable and widely used Flash design and development tools are introduced along with links to further information. Additionally, developers may want to have a look at a recent article by InfoQ titled "The State of Flex RIA Development Ecosystem"[130] which provides a comprehensive summary of many popular IDEs, frameworks, and other tools currently available for Flex developers.

Flash tooling can be generally categorized as either 'developer' or 'designer' oriented. Developer tools tend to be more code oriented (e.g., programming IDEs), while designer based 'authoring' tools tend to use visual oriented workflows (e.g., GUI drawing and animation tools). Designers will often speak of creating 'content' and 'media', while developers code or develop 'applications' or 'programs'. As well, the software built by developers tends to be data-driven applications, often utilizing client-server style architecture. In reality the designer and developer groupings are not absolute, both the tools themselves and their users can, and often do, overlap. In fact, one of the unique aspects of the Flash platform is it often brings together both the artistic designers as well as the logic based programmer communities. Flash tooling increasingly enforces this hybrid of form and function, and has proven beneficial in many client facing applications. Whether it is building games, data visualizations, web applications, or other rich internet applications, the creation process for Flash software is often both a science and an art.

For designers, Flash applications and content are typically created using an authoring tool and saved as Flash (.fla) and ActionScript (.as) source files. Developers on the other hand will often build applications using the Flex framework and mxml UI markup language inside of a code-centric IDE. Within these Flex coding environments developers can write and save code in mxml, ActionScript, or both. Regardless of the development tools and source file format the end result after compiling is generally the same – a .swf for execution by the Flash player.

4.2 Adobe Tools

While portions of the Flash platform have been opened up to the public domain, Adobe continues to maintain significant portions of ownership, especially with regard to the Flash runtimes and select tooling. Adobe's business model has always centered on its software tools, and this is no different for Flash. Adobe's commercial design and development tools for the Flash platform continue to be among the most recognizable and popular. They offer a broad suite of highly integrated tools for creating Flash software. The core of this Flash development ecosystem is Adobe's Flash Professional and Flash Builder (formerly Flex Builder). To a lesser extent, Adobe's newest Flash 'prototyping' tool Flash Catalyst can be included as well. The tight integration of these core Flash tools with Adobe's remaining suite of designer tools like

Illustrator, Dreamweaver, and Photoshop means simpler workflows between designers and developers alike. For example, visual assets can easily be created in Illustrator and imported natively into Flash Professional. Adobe's Flash Catalyst product aims to do this workflow one better by allowing those same visual assets to be converted to functioning user interface components with the click of a button.

Until recently, the end result of any Adobe Flash tool chain was essentially always the same – a Flash application in the format of a swf file. The swf file might be compiled for a specific version of the Flash player, like older versions for compatibility, or alternatively compiled for the lightweight Flash Lite player. The swf might also be embedded inside of an html file (e.g., web application), or bundled inside an installable AIR file (e.g., desktop application). Again, regardless of the development tool and deployment model used, the result was always a swf file that is only executable using the Flash player. This 'Write-Once-Run-Anywhere' (WORA) software development model has, until recently, been fundamental to the Flash platform or any software which executes inside a virtual machine. However, Adobe is introducing an exception to the WORA process specifically for Apple's iPhone with the newest versions of Flash Professional and Flash Builder. Developer's using these tools will compile their 'Flash' iPhone applications specifically for the native iPhone OS, and not as swf files for the Flash player. This exclusive 'Write-Once-Compile-Anywhere' (WOCA) process introduced for the iPhone does run counter to the fundamental goals for the Flash platform outlined in Adobe's Open Screen Project. More specifically, a WOCA style development model circumvents the goal of having the same consistent application runtime across ALL devices – the Flash player. Obviously this exception is to compensate for the lack of a Flash player on the iPhone, the last holdout in terms of smartphones without the Flash player.

4.2.1 Flash Professional

The grand-daddy tool for creating Flash content (and arguably the most popular) is the "Flash authoring tool", or more specifically Adobe's Flash Professional, currently at version 'CS5'. The "CS5" signifies it is a constituent of Adobe's Creative Suite family of software design products. Adobe markets Flash Professional as a commercial product available for Mac and Windows based systems.

Flash Professional is a powerful integrated development environment (IDE) for creating, modifying, compiling, testing, and managing Flash content. Like many modern IDEs, the workspace is highly customizable using several toolbars and panels. Figure 85 shows a typical layout view inside the Flash Professional CS4 authoring environment. The tool is designer (visual) oriented, and uses a timeline development paradigm which reflects its roots as a web graphic animation tool. The graphical construction area within the Flash authoring tool is a 2D visual editing canvas referred to as the stage. Workflow generally centers on creating, animating, and adding interactivity to graphical assets referred to as symbols. Basically applications are constructed using symbols, components, and ActionScript code. Graphics and animation can be made at design time using Flash Professional's WYSIWYG drawing and animation tools, or created on-the-fly at runtime using ActionScript. Unlike drawing and animation in Flash Professional, interactivity is accomplished exclusively using ActionScript.

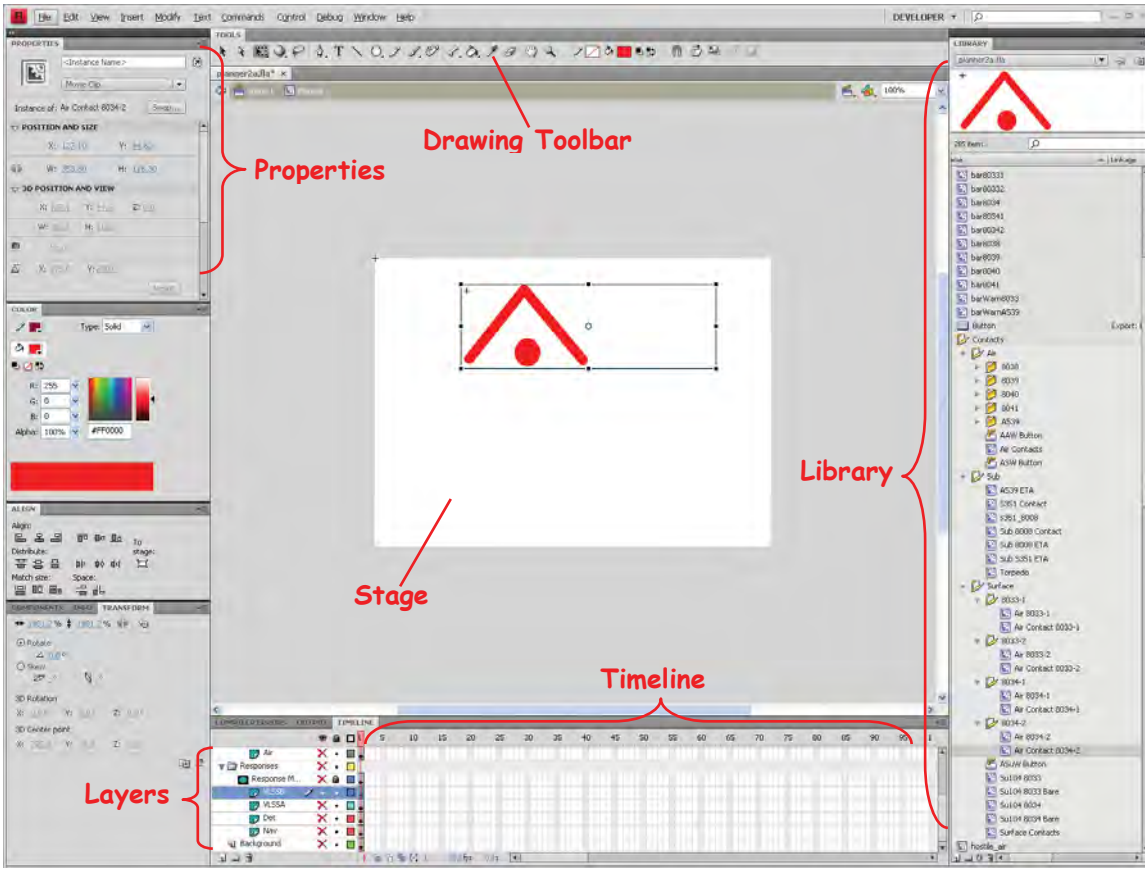

Figure 85: Adobe's Flash Professional CS4 is a designer focussed authoring tool with heavy emphasis on creating and manipulating visual assets.

4.2.1.1 Layers

Drawing on the stage follows a layer (z-depth) paradigm in which the display objects belonging to higher layers obscure those belonging to lower levels. Within the Flash IDE there are 4 types of layers: *drawing*; *motion guide*; *mask*; and *guide* layers. Guide layers are strictly a design-time tool to aid in the layout and alignment of graphic assets. Anything placed on a Guide layer is completely ignored during the compile process. Motion guide layers are used to animate graphics along a specific path of motion. Layers with a masking layer applied are completely obscured from view at runtime except for areas covered by graphics placed on the masking layer. For example, a newly masked layer will be completely invisible until a filled shape is placed on its masking layer. At this point everything directly 'underneath' the masking layer shape(s) will become visible (assuming the shape's transparency is off). Note drawing layers themselves are strictly an aid for the developer at design-time, and are not included in the compiled runtime files.

4.2.1.2 Frames

Within the Flash authoring tool, content is temporally organized by frames arranged sequentially along horizontal timelines. Each individual frame corresponds to an instance of the stage at a specific point in time. For example a button or symbol instance is located at a particular spot on the stage at a particular point in time (its frame). Content added to the stage by the developer must be done inside of explicitly defined "keyframes". The process of automatically generating content to fill frames between keyframes is known as motion and/or shape tweening.

4.2.1.3 Components

Components are small pre-built reusable widgets that provide common yet specific functionality. The Flash authoring environment provides a number of pre-installed components, mainly in the form of UI controls. Third-party components (.mxp files) are available from numerous online sources (e.g., Adobe's Flash Exchange) and can be installed using Adobe's free Extension Manager tool. Although components are pre-compiled and generally not editable, they often provide a number of parameters which can be set at either design-time or run-time to control a component's look and behaviour. Refer to Section 5.1.2 for more information regarding Flash components and extensions.

4.2.1.4 Symbols

Symbols are to Flash authoring what classes are to object oriented programming; they are "instantiated" either at design time (e.g., whenever a symbol is dragged onto the stage), or at runtime using ActionScript code. More specifically, symbols are reusable Flash visual assets which are further decomposable into constituent symbols, components, and shapes. Each symbol has its own timeline which plays back independent of the parent or main timeline. Symbols are managed using the Library component within the Flash IDE and can be organized in a folder tree hierarchy manner similar to layers. Symbols are needed for every visual asset which needs to be animated and/or manipulated programmatically. Generally speaking, more complex symbols incorporate more animation and interaction, and therefore typically contain more sub-elements and more ActionScript code.

Symbols can exist as one of three types: *graphic*; *button*, or *movie clip*. Graphic symbols are static images or graphics which have no animation or interactivity. Button symbols generally do not have animation but are interactive. ActionScript can be either attached to the button symbol and/or embedded inside separate frames for the *Normal*, *Over*, *Down*, and *Hit* button states. Movie clips are similar to graphic symbols but can also have interactivity using ActionScript. They range from simple graphic sprites to complex compilations of nested symbols and components complete with animations and embedded ActionScript code.

4.2.1.5 ActionScript Code

As mentioned, ActionScript is the built-in scripting and programming language of the Flash platform. It is a powerful object-oriented programming (OOP) language extremely similar to Java. In general, all ActionScript code is either embedded within the Flash source files (.fla) or stored separately in external ActionScript source files (.as). Within Flash source files,

ActionScript code can be embedded in a number of places. More specifically, it can be attached to any frame(s) within the main timeline, any frame(s) within a nested movie clip, and/or attached directly[3] to individual instances of symbols and components on the stage. By convention, ActionScript code attached to the timeline is typically embedded within the first frame of a separate (top) drawing layer labelled "actions". Frames containing ActionScript code are indicated by a lower case "a" embedded in the upper half of the frame's block symbol on the timeline.

4.2.2 Flash Builder (formerly Flex Builder)

Flash Builder 4 is Adobe's developer-oriented IDE for building rich internet applications using the Flex SDK. The Flex SDK by itself is a freely available open source tool, however the Flash Builder IDE is a commercially licensed product based on eclipse[4]. It can be installed as an eclipse plug-in or standalone application. As mentioned previously, the current version 4 represents a renaming of the tool, as all previous versions were named 'Flex Builder'. To further enforce the confusion, the underlying SDK used by Flash Builder 4 retains the Flex moniker.

Flash Builder is available for both Mac and Windows operating systems in two versions – 'Standard' and 'Premium'. The two editions are essentially identical, however the Premium edition adds a handful of advanced features related to performance testing (e.g., memory and CPU profiling), as well as support for automated testing. The Adobe website provides a features comparison table[131] to help distinguish between the two Flash Builder versions. Note the standard version is available free to students or with the purchase of Flash Professional CS5.

Typically the applications built using Flash Builder are data driven and more functionally oriented, in contrast to the heavily themed nature of traditional creative Flash content. For programmers already experienced with eclipse or any other modern IDE, Flash Builder will feel very familiar (Figure 86). The IDE uses the standard panel based layout (perspective) which can be customized as desired, or developers can simply use one of the default arrangements. The main center panel is normally used for the source and design views which can be toggled using their respective tab buttons.

[3] Only ActionScript 1 and 2 can be attached directly to symbols, and this option was removed with ActionScript 3.
[4] Eclipse is a popular open source IDE which uses plug-ins for extensibility.

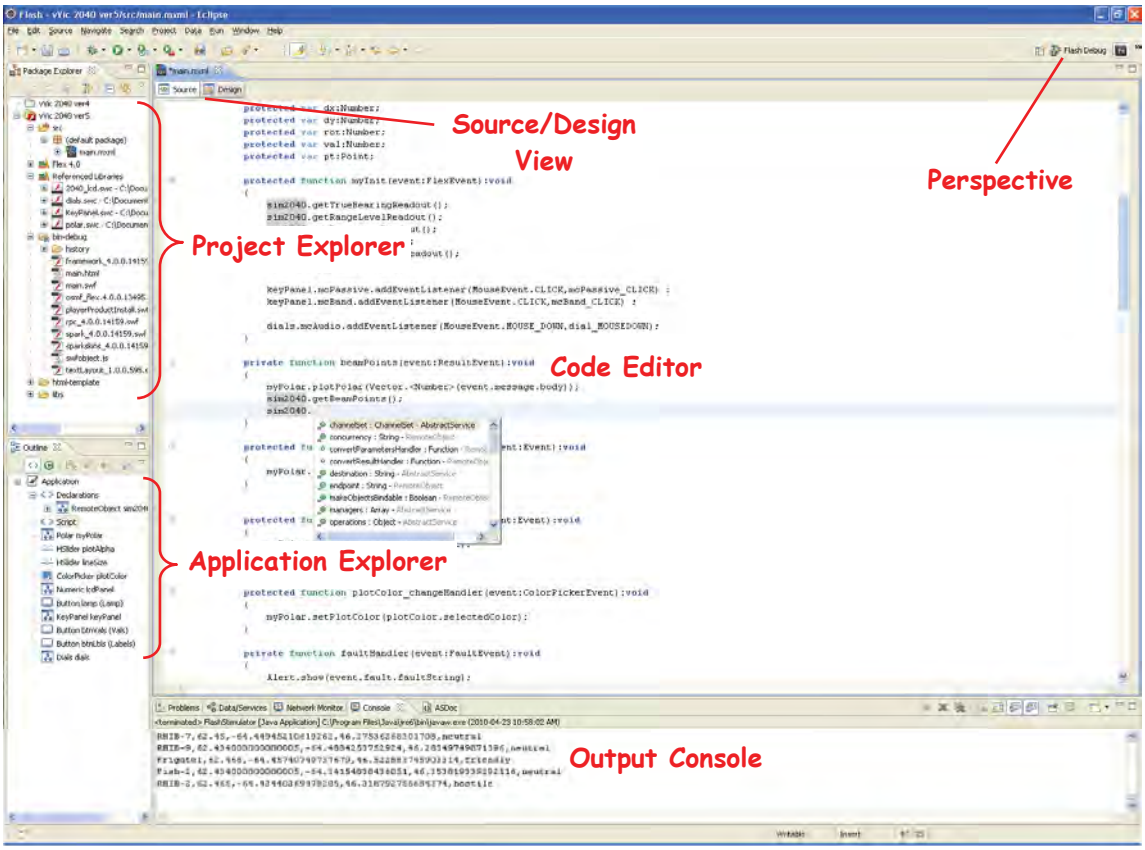

Figure 86: Flash Builder 4 is Adobe's eclipse based IDE for developing Flex applications.

The source view editor enables developers to write, edit, and review both mxml and ActionScript code. Standard coding features include refactoring, code hinting, block commenting, and code folding. The design view provides many graphical components with drag and drop support for visually laying out application GUIs (Figure 87), while automatically generating the underlying mxml code. Other noteworthy features include interactive debugging, web services description language (wsdl) introspection, and wizards for back-end data connectivity.

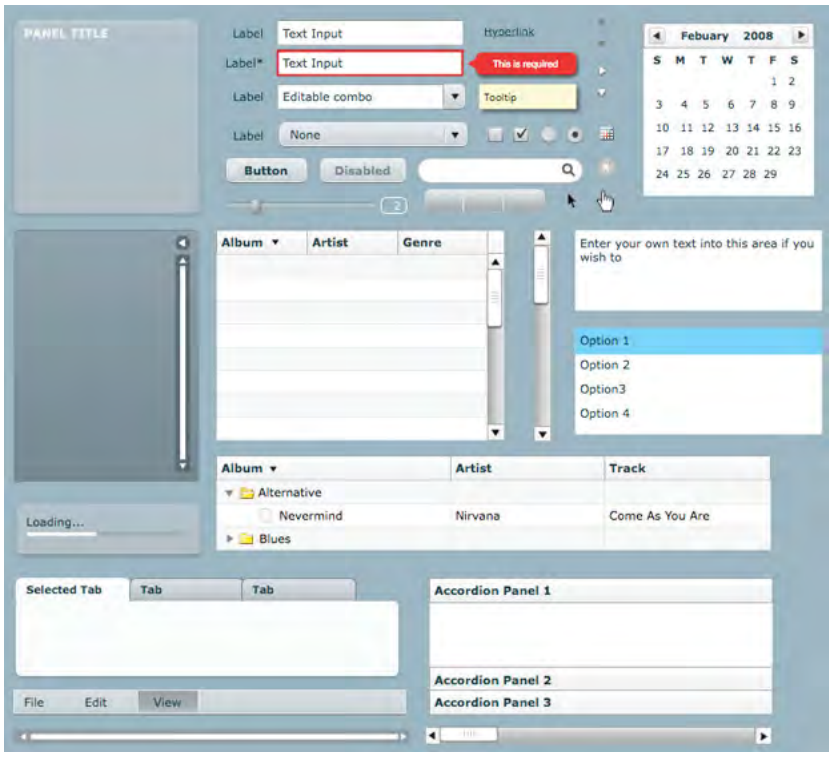

Figure 87: The Flash Builder IDE includes several UI components by default.

4.2.3 Flash and Flex Workflow

Both Adobe's Flash and Flex development tools have their respective strengths and weaknesses as mentioned previously. As Flash projects can often incorporate significant amounts of both visual design as well as programmatic functionality, it can be less than clear what the best tool for the job is. Derrick Grigg provides (via his blog) a nice summary of factors to consider when choosing between a designer tool like Flash Professional and a developer IDE like Flash Builder[132].

When to Flex:

- *mid to large applications*
- *data heavy*
- *charting, graphing data visualization requirements*
- *modular development*
- *lots of form input UI screens*
- *not as concerned about final SWF size*

When to Flash:

- *marketing, promotional, interactive sites*
- *design heavy*
- *highly customized UI*
- *online games*
- *animation*
- *concerned about final SWF size.*

4.2.3.1 Using Flash Assets in Flash Builder

Of course it can often be advantageous and even necessary to use both tools when building a new application. There are three basic options for incorporating Flash Professional assets into a Flex application – **embed**, **load**, or **compile**.

To embed an asset, simply use MXML's embed tag:

```
[Embed(source="assets/target.swf")]
[Embed(source="assets/library.swf", symbol="Contact")]
```

This assumes the swf asset file is located in a root subfolder called 'assets'. The latter specifically embeds the "Contact" symbol from the "library.swf" file. Embedded assets are compiled into the swf file of your Flex application. They are not loaded at run time, and you do not have to deploy the original asset files with your application. However, you cannot directly or easily access the properties or methods of embedded SWF files. This approach works best with simple assets that have relatively small file sizes.

A second option is to load Flash assets at runtime using the Flex SWFloader component:

```
<mx:SWFLoader id="ntdSwf" source="../assets/ntd_symbols.swf" autoLoad="true" />
```

Because the Flash assets are not compiled into the Flex application, they are stored separately and therefore must be deployed with the Flex application file for runtime loading. This method is great for simple content like audio and video with larger file sizes.

A third option is to export (publish) the asset from Flash Professional as a Flex component (swc file), and subsequently import the custom component to any Flex project. Note this workflow requires the pre-installation of the Adobe Flex Component Kit. This kit is a free extension for Flash Professional which enables any library assets within the authoring IDE to be published as custom Flex components.

To subsequently use the custom component, simply add the component's swc file to the build path of the target Flex project. More specifically, within Flash Builder open the project's properties, go to 'Build Path' > 'Library Path' and add the custom swc file created using Flash Professional. Now in Flash Builder you can use the ActionScript import statement to enable access to the class and all its members. This approach gives full access to all the methods and

properties of the Flash asset at design time. Like all Flash Builder library classes, assets in a swc are loaded and included at compile time. Remember to use the Flex UIComponent as a container when adding graphical assets to the Flex application. This option is most popular for leveraging more complex Flash assets with their own APIs.

4.2.4 Flash Catalyst

It is not hard to understand the motivation behind Adobe's newest designer tool - Flash Catalyst. Starting with nothing more than artwork of a proposed application, import the visual design into a tool that 'automagically' transforms it into a functioning application…, or at least a working prototype. Catalyst is targeted towards visual artists, interaction designers, and other non-programmers who are uncomfortable writing code and using developer tools. It essentially gives designers the ability to iteratively experiment with both the look and **feel** of a new application, and subsequently leverage these early design efforts in the final development of the application.

The Catalyst version of workflow for creating Flash based RIAs (Figure 88) begins with a design proposal created using tools like Illustrator, Photoshop, and Fireworks. The visual assets are then imported into Flash Catalyst and converted, with 'minimal' effort, into functioning components. Catalyst also makes it easy to define 'states' for both the application (e.g., logged in versus logged out) and its components (e.g., enabled versus disabled). At this point the design is a tangible interactive application, and for smaller lightweight applications that are not data driven, it may even be deployed as a finished swf file. However, for most applications the final step is still left to developers, who import the Flash Catalyst project into Flash Builder for completion. The remaining work includes replacing the design-time sample data with real data by connecting it to a server, and customizing component behaviours as needed.

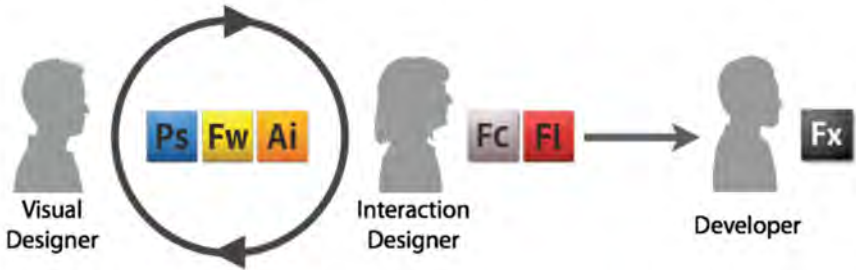

Figure 88: Adobe's version of workflow for creating Flash based RIAs uses Catalyst to bridge the gap between early design artwork and finished applications [133].

Adobe lists Catalyst's key features as:

- *Transform artwork created in Adobe Photoshop®, Adobe Illustrator®, and Adobe Fireworks® into interactive designs.*
- *Rapidly create interactive prototypes with the ability to leverage them in the final product*
- *Publish a finished project as a SWF file ready for distribution*
- *Work more efficiently with developers who use Adobe Flash Builder™ 4 to create rich Internet applications (RIAs). Designers use Flash Catalyst to create the functional user*

experience then provide the project file to developers who use Flash Builder to add functionality and integrate with servers and services. [134]

Figure 89: Flash Catalyst provides a designer friendly visual GUI for converting artwork into functioning components (image from cnet[135]).

As of this writing Adobe have no official release date for Flash Catalyst, however the current beta 2 version is available via free download from Adobe's web site.

4.2.5 LiveCycle Enterprise Suite 2

While not entirely relevant within a Defence research context, Adobe's LiveCycle Enterprise Suite 2 (ES2) is nonetheless worthy of mentioning. LiveCycle ES2 is Adobe's solution (Figure 90) for automating business processes for large corporate and government organizations. "It combines technologies for data capture, information assurance, document output, content services, and process management to deliver solutions such as account opening, services and benefits enrolment, correspondence management, request for proposal processes, and other manual based workflows"[136]. These processes are generally document based (e.g., pdf) user-centric workflows which often involve individuals both internal and external (e.g., customers, partners, citizens) to an organization. A simple example might be a customer completing an interactive form for a loan application which is then submitted and routed through a bank's internal review and approval process.

Figure 90: LiveCycle ES2 is Adobe's business automation solution for large corporate and government organizations which leverages Flash clients, tools, and server-side technologies (image from [137]).

Architecturally LiveCycle ES2 taps into many components of Adobe's pdf and Flash platforms to enable streamlined and centralized digital alternatives to existing manual and paper based business processes. Not surprisingly, backend data services for LiveCycle ES2 applications (Figure 91) are provided by Adobe's LiveCycle DS data services product. In addition to Flash Builder, tooling includes 'Designer' for pdf form design and 'Workbench' for business process modeling.

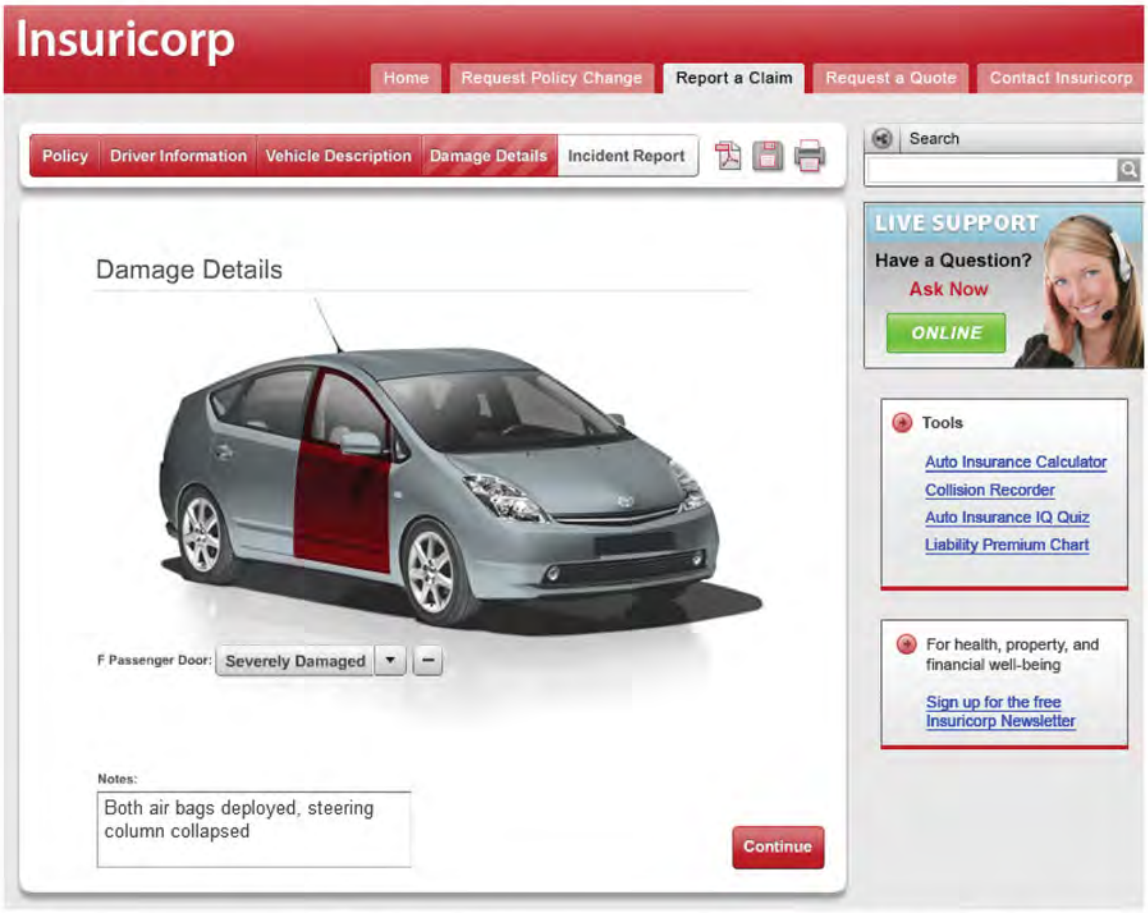

Figure 91: An example of a RIA for submitting insurance claims powered by LCDS ES2. The application features wizard-style guided navigation for self service customers, real-time document collaboration, and a live chat option. Being pdf based, "the guides can also be taken offline, allowing users to interrupt the process and continue later at their own convenience. And users can save fully completed PDF forms for their records after the process is completed"[138].

4.3 Third-Party Flash Tools

It is a common misbelief among many non-technical people that Flash is an exclusive software platform of Adobe. While Adobe does provide the most popular versions of the Flash runtime clients, and to a lesser extent Flash development tools, in truth there are countless Flash development tools from both commercial and open source channels.

For example several third party Flash IDEs exist in both commercial and free variations. For significant Flash content design (e.g., timeline animation, advanced graphics, visual effects, masking, etc), Adobe tools such as Flash CS4 Professional enjoy very little competition. This dominance within the design space is further enforced by the tight integration between Flash

Professional CS4 and Adobe's supporting suite of designer tools such as Illustrator, Dreamweaver, and Photoshop. At present, searching for alternatives to Flash Professional reveals very few tools with similar feature sets. A short list of similar commercial offerings might include products such as Sprout Builder (Section 4.3.3) and SWiSH Max 3. Currently, no open source tools appear to exist that are directly comparable to Flash Professional. Conversely, when it comes to Flash application development (e.g., coding, compiling, GUI layout, debugging, and previewing), several third party Flash IDEs exist in both commercial and free varieties. It is important to note these IDEs are developer oriented code-centric tools for working with Flash languages like ActionScript and mxml. A few of the more popular tools for creating Flash content and applications are summarized below.

4.3.1 FDT

Arguably one of the best IDEs for developing with ActionScript, Powerflasher Solutions' FDT (Flash Developer Tool) is a commercial Flash IDE available as an eclipse based plug-in for both Windows and Mac OS (Figure 92). The company describes FDT as a "powerful development environment for Flash coding, MXML, and AS2/AS3 programming"[139].

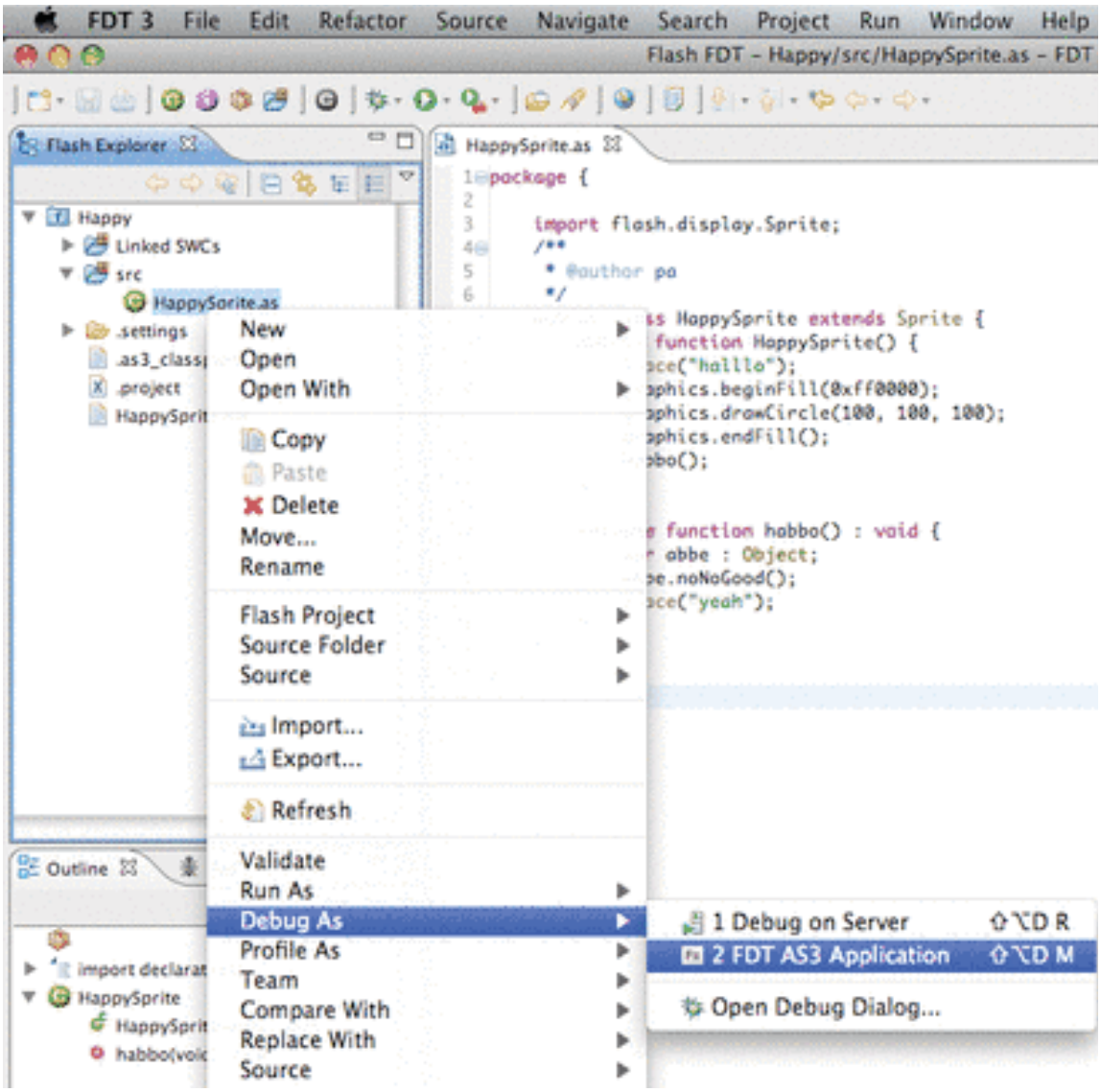

Figure 92: FDT is a popular commercial product for developing Flash applications available as a plug-in for the open source eclipse IDE.

FDT and Flash Builder are two very similar commercial offerings with similar feature sets for developing Flash applications. Comparing the two is tricky as both are moving targets in terms of their features, support, and price. Historically FDT has received slightly more favourable reviews[140-141], however it is typically pricier and does not have a user community any where near the size of Adobe's Flash Builder.

4.3.2 wonderfl

Wonderfl (Figure 93) is a completely online and free Actionscript3 editor. Wonderfl represents yet another example of a growing number of 'Software as a Service' (SaaS) applications. From the wonderfl website:

> *Usually when you want to build Flash, you need Flash IDE or Flex, FlashDevelop or any flash building tool to write code and compile it. With wonderfl, you write Actionscript3 code in the html form, and our server compiles your code. You'll see warnings, maybe errors from the compiler, and if your code is valid, your Flash(swf) will be automatically shown besides your code. Write code and watch it in action, realtime, in your browser*[142].

A key advantage to this type of online editor is that it is platform agnostic; you can code Flash applications from any OS with a web browser. Furthermore, hardware resources such as storage and CPU are irrelevant; you can compile and run large and/or complex programs using the lowest of entry level nettops.

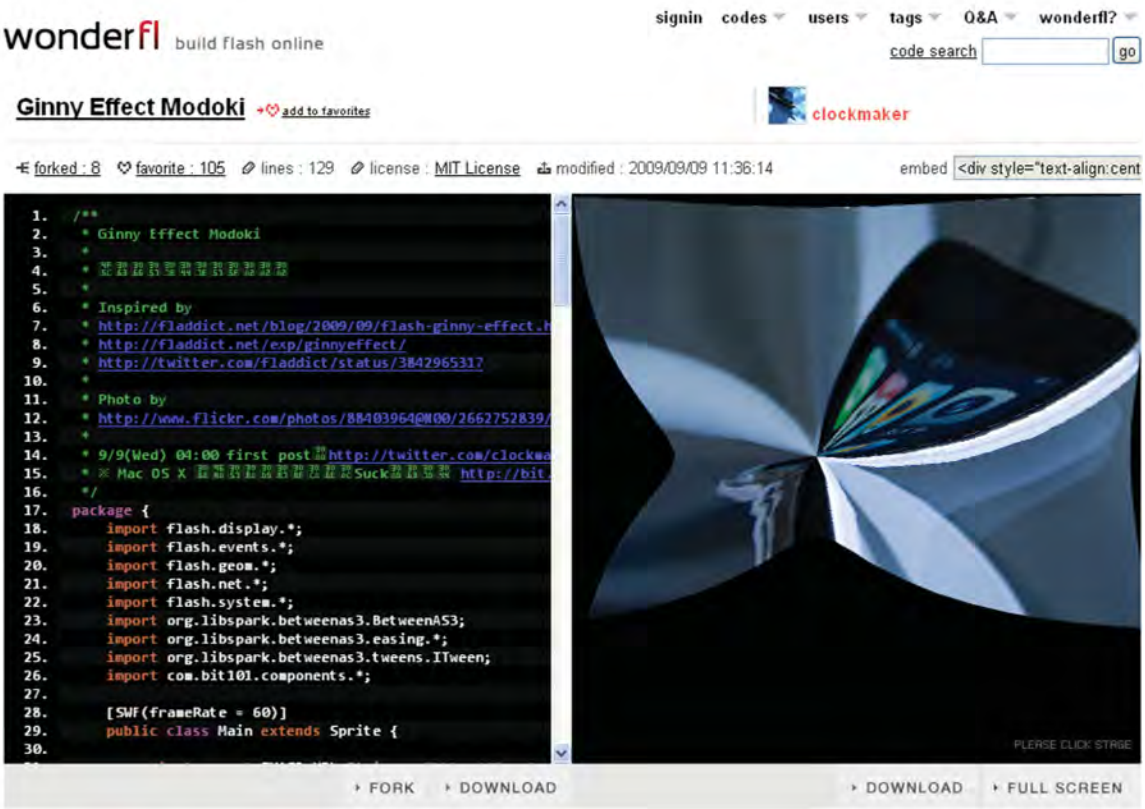

Figure 93: Using the wonderfl online AS3 editor to write Flash programs. The editable source code is displayed in the left panel and the compiled running swf file in the right panel.

4.3.3 Sprout

Sprout is a completely online (browser-based) WYSIWYG tool for creating Flash content, which itself happens to be a Flex based RIA. In contrast to wonderfl's code focused editor for developers, Sprout's drag-and-drop style GUI appeals more to Flash designers working with visual assets. "Designers can use it to create, publish and track Flash widgets, websites and mashups"[143].

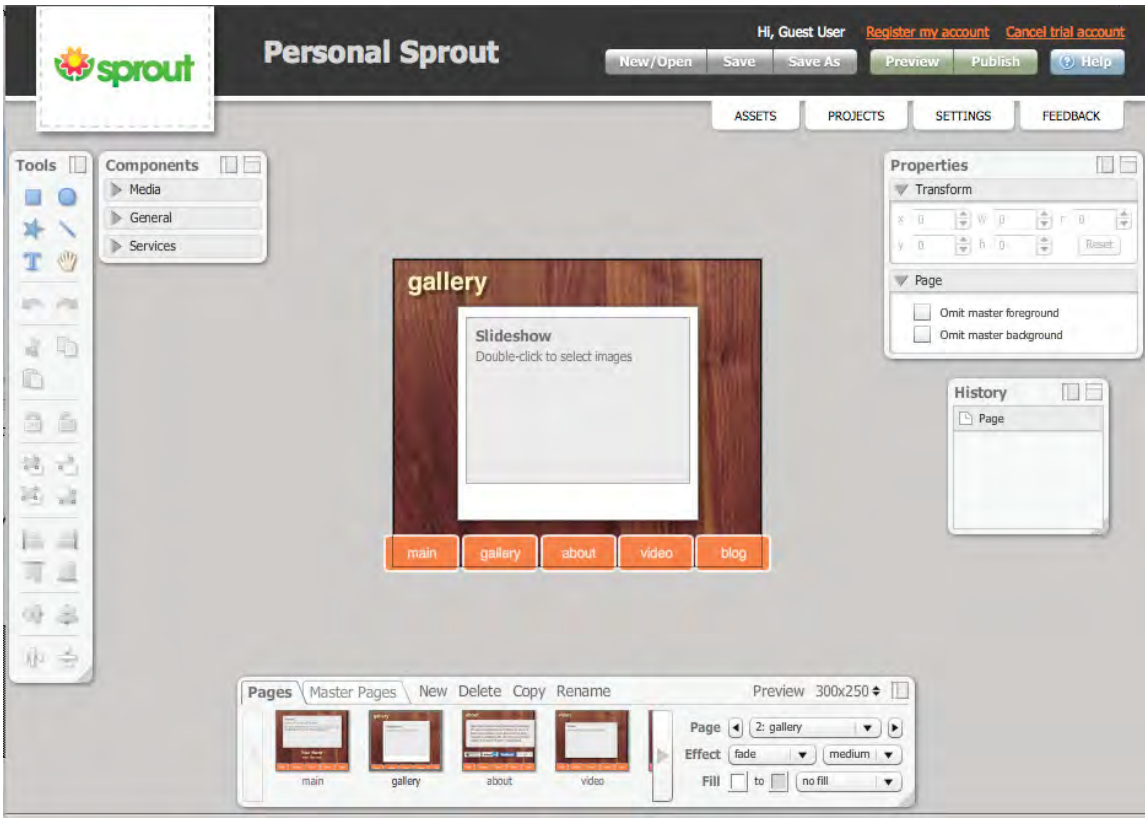

Figure 94: Sprout is a completely online (browser-based) WYSIWYG editor for Flash built using Flex (Image from [144]).

4.3.4 Visual Studio Developers

SapphireSteel Software offer the Amethyst IDE[145] plug-in for Microsoft's Visual Studio. Amethyst is a full featured IDE extension for Microsoft Visual Studio for creating Flex and AIR applications. The Amethyst IDE can be installed into the commercial editions of Visual Studio or the free Visual Studio shell. Amethyst offers a familiar development environment for the huge community of Visual Studio developers.

Ensemble Tofino[146] is another Flex plug-in for Visual Studio which enables .NET developers to create Flex and AIR applications. Tofino is free and includes numerous coding features such as full debugging, object/class browser, build wizards, and code completion.

4.3.5 Java Developers

Generally speaking Java developers will find many of the Flex developer oriented tools quite familiar. This is especially true for tools such as Adobe's Flash Builder and Powerflasher Solutions' FDT, as they are based on the widely popular eclipse IDE already used by a majority of java developers. Also of note for java developers is JetBrain's IntelliJ IDEA, a popular IDE for java developers which features integrated support for coding Flex projects. The ultimate edition of IntelliJ IDEA is free for open source projects (commercial work requires a paid license) and its features include Flex code editing, code completion, and code refactoring.

4.4 Open Source Flash Tools

Generally speaking, open source alternatives exist for pretty much any component of the Flash platform, including runtime clients, developer tools, and server side technologies. OSFlash is a website dedicated to all things related to open source Flash and hosts an extensive collection of relevant projects. A few noteworthy open source Flash tools are briefly presented below in this section.

4.4.1 FlashDevelop

FlashDevelop[147] is a relatively mature and feature rich open source IDE available for the Flash platform. The editor is supported by a small but active community of users and contributors. The editor is Windows based, however according to the FlashDevelop website, it is compatible with Mac OSX and Linux using virtualization software.

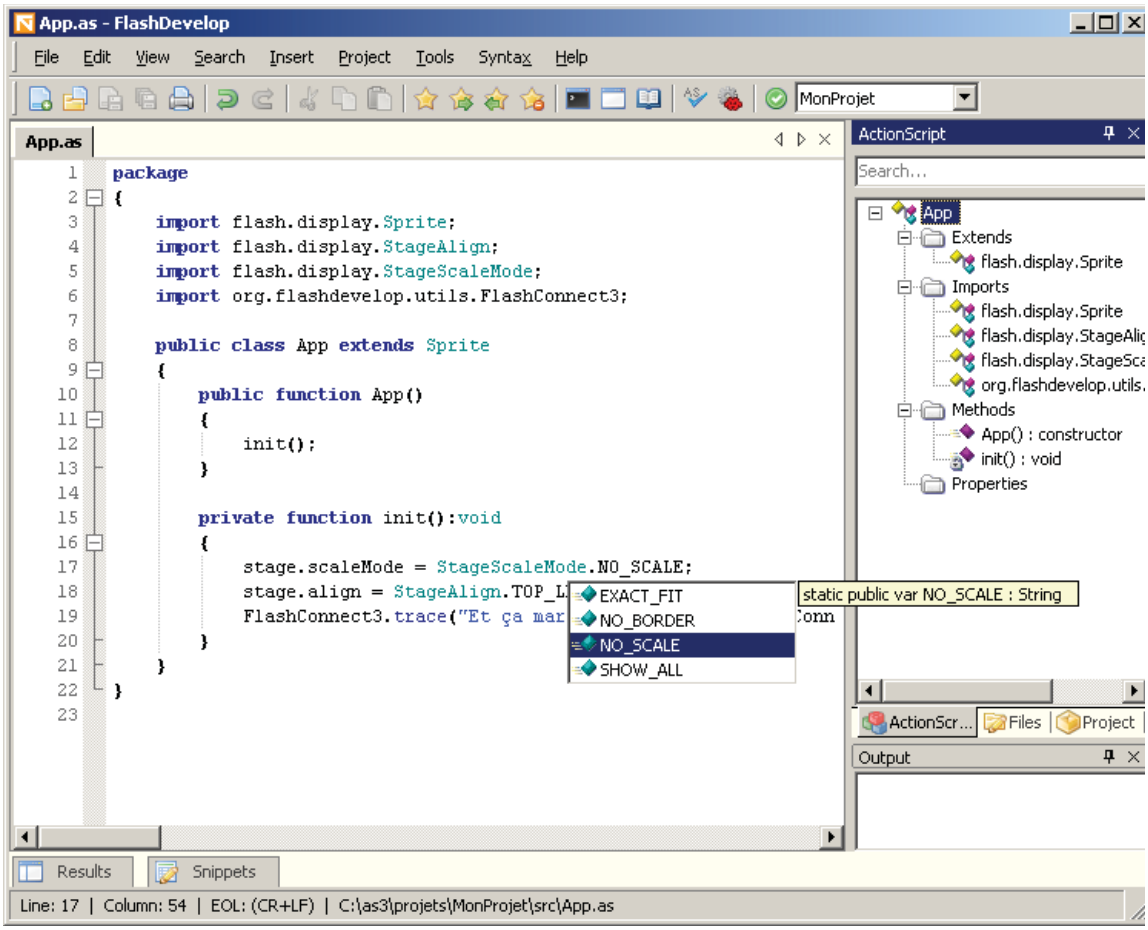

Figure 95: The FlashDevelop IDE is a free open source Flash IDE[147].

Key features offered by FlashDevelop include[147]:

Supports three languages: ActionScript 2, ActionScript 3 & MXML and HaXe[5].

Brilliant Actionscript Code Completion (IntelliSense) and code generators.

Code completion for XML, MXML and HTML

Code navigation (jump to declaration), just pressing F4/Shift-F4.

Easy integration with Flash and command line compilers, just using Ctrl+Enter to compile.

Integrated Project Manager to handle all the project assets, properties and files.

Smart project templates to quickly get started.

Instant swf building with MTASC or MXMLC using custom comment tag @mtasc/ @mxmlc.

Easy to use Context API search.

[5] HaXe is an open source multi-platform programming language. HaXe programs can be compiled into Javascript (.js), Flash (.swf), and PHP (.php) executables.

Customized GUI for AS2API & ASDocs documentation generators.

Lionel Low provides a good feature-by-feature comparison between FlashDevelop and FDT on his blog[148]. Both editors trade strengths and weaknesses however FDT generally offers more advanced coding features. At the same time FDT tends to also have a steeper learning curve. Either way, for Lionel FlashDevelop ultimately has the trump advantage of being free.

4.4.2 MiniBuilder

MiniBuilder[149] is a free open source IDE for AS3 with a unique twist, the IDE itself is written in AS3. The project is hosted on Google code and the developers plan to offer the IDE in both a desktop (AIR) and browser (server side) version. Currently at this early stage only the desktop version is available as an alpha, while the web version is planned for early 2010. Some of the unique advantages include its lightweight (under 3Mb) and runs on both Linux and Windows (Mac OS is planned). The developers hope to eventually run MiniBuilder on smaller hardware platforms like netbooks or future pocket-sized devices.

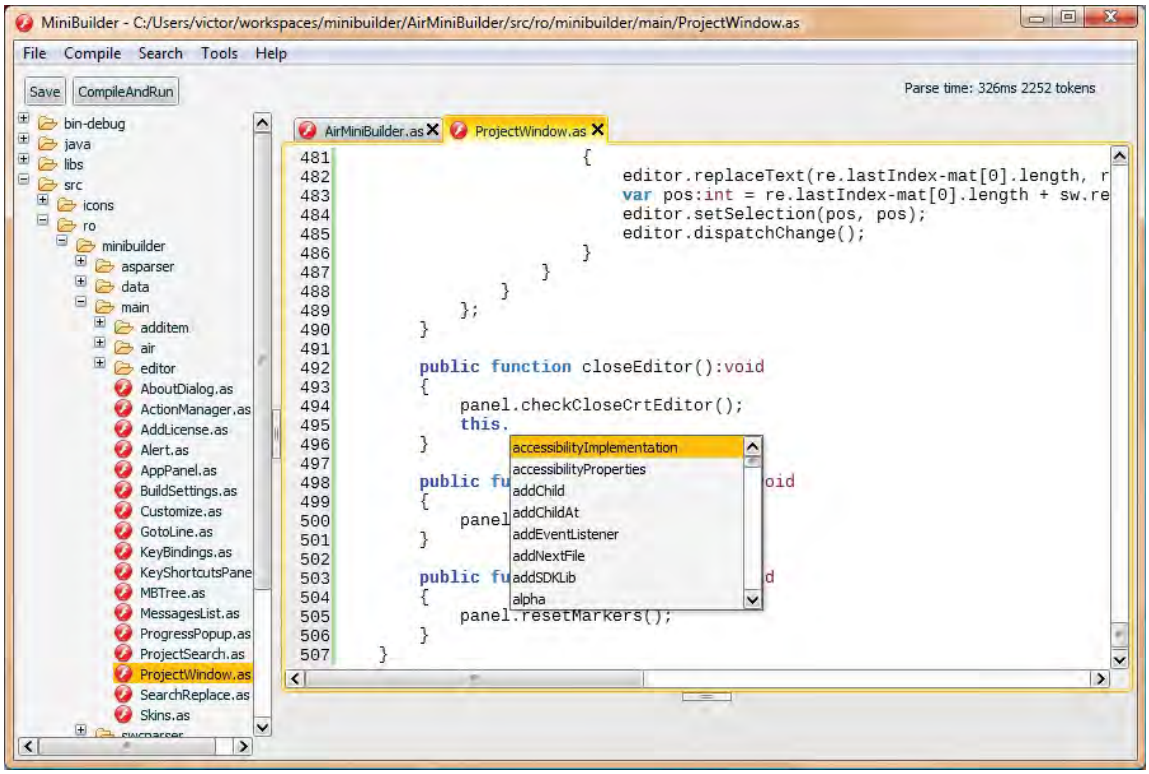

Figure 96: MiniBuilder is an online open source IDE for ActionScript created using ActionScript.

4.5 Alchemy

A final noteworthy item in considering tooling for the flash platform is Alchemy; a set of developer tools for creating Flash applications from C and C++ source code. More specifically, "the Alchemy tool chain includes a set of tools for building, testing and debugging C/C++ projects, example projects and documentation"[150]. So why would a Flash programmer want to use any language other than ActionScript or mxml for creating Flash applications? More specifically, why C or C++? Essentially there are two key motivations developers might have for using C/C++ code. Firstly, consider there are millions of lines of pre-existing free and open source code written in C and C++. Developers can save time by re-using code from the massive libraries of C and C++ which already exist. The second reason is simply the improved speed of many specific functions written in C/C++.

5 Extendability

5.1.1 SWCs and APIs

ActionScript is a general purpose object-oriented programming (OOP) language. Like most such modern OOP languages, one of its core strengths comes from how easily it facilitates the sharing and reuse of code. Developers can easily create and share libraries of programming assets (typically in the form of class files) for specific yet common functionality. A simple example might be a postal code validator class which ensures a text entry is a valid postal or zip code.

The standard means by which developers organize and share reusable code is to group classes into packages and packages into libraries. Typically these libraries and other source assets are then bundled together and compressed into a single archive file. In Flash's case these API archive (aka component) files are referred to as "swc" (pronounced "swick") files. For java developers, swc files are the Flash equivalent of jar files.

> *A SWC file is an archive file for Flex components. SWC files make it easy to exchange components among Flex developers. You need only exchange a single file, rather than several MXML and ActionScript files, images, and other resource files. Also, the SWF file inside a SWC file is compiled, which means that the code is obfuscated from casual view.* [151]

Larger collections of source libraries are often referred to as an "API" (application programming interface), and occasionally the terms "engine" and "framework" are also ambiguously used. A software development kit (SDK) generally refers to a broader API tool set which usually includes a compiler, debugger, and other developer tools. In any case, these APIs usually focus on a specific area of functionality, like for example a 3D API or physics API. Many if not most APIs are open source projects available through a website and often supported by an online community of developers who share a common interest in the capability exposed by the API. In the case of Flash there are APIs for everything from sound, video, text, to animation, database operations, social networking, augmented reality, multi-touch, and countless more.

Generally speaking there are thousands of Flash specific APIs available across the web for developers, whether through open source or commercial sources. Probably the easiest means for finding specific libraries is to simply use a general search engine such as Google. Developers can also search specific code hosting sites like SourceForge, github, and Google Code. Regardless of the search tool, suggested keywords which can be used include "Flex", "Flash", "ActionScript", "AS3", "MXML", and "SWC" along with more generic source code terms like "API", "engine", "framework", "library" and "SDK". The developer area of Adobe's website is another great source for many developer files and tools. While by no means exhaustive, the Resources portion of this document (Section 8.6) contains a compiled list of links to some more recent and useful API collections complete with summarized descriptions.

5.1.2 Extensions (aka Components)

Adobe Exchange is a file sharing area on their website where users can "download extensions, custom tags, scripts, content, and other items that extend the functionality of Adobe applications"[152]. Generally these files add or enhance specific functionality to using Adobe software and can often save design and/or development time. About the only other commonality among all these files is they typically require little or no programming to install and use, as opposed to more developer oriented tools and files such as libraries, APIs, SDKs, and frameworks.

There are thousands of items available to pick from and the exchange area is organized according to Adobe product (e.g., Flash Exchange section). "Each extension has its own page that includes the download links, a short description, user ratings and reviews, and a link to the respective online discussion forum where you can post questions and get support for that extension"[152].

Figure 97: Adobe exchange offers numerous free and commercial extensions for Adobe software.

Many extensions are bundled and offered using the extension package (MXP) file format: "a compressed installation file containing the extension(s) to be downloaded and installed by the Adobe Extension Manager". Importantly, files utilizing the MXP format can ONLY be installed using the Adobe Extension Manager[153], and only installed for the specific Adobe application they were written for. Extension Manager is a small free tool for installing and managing extensions for Adobe software.

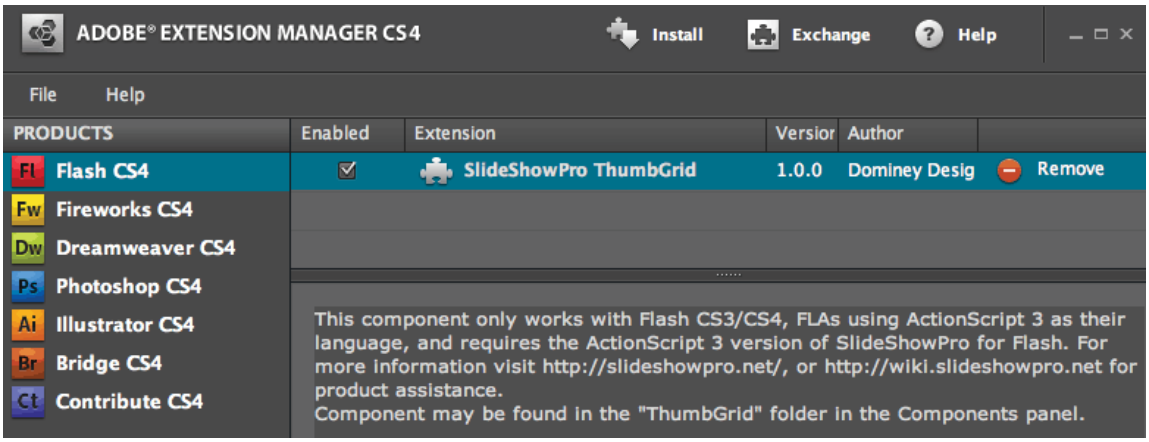

Figure 98: Adobe Extension Manager is a free and required utility for installing Adobe software extensions that are packaged using Adobe's proprietary MXP format.

Of course Adobe is not the only place to find large selections of components, templates, scripts, and other reusable assets to aid Flash developers, there are numerous web sites offering both free and commercial files for download. The selection of files is often very diverse, but some of the common categories include "Galleries and Slide Shows", "Menus and Navigation", "Effects", "User Interface", "Sound", "Video", and "Animation". As a quick start reference, a few popular sites (in no particular order) include:

- Advanced Flash Components[154] – Use MXP format exclusively; home of UMAP component (very impressive mapping component);
- Flex.org[155] – great selection of free and commercial Flex specific components;
- RIAForge[156] – exclusively open source components for all Adobe products including Flash Builder, Flex framework, and Flash;
- ActiveDen[157] – huge and varied selection of Flash and Flex files for purchase only; formerly known as "FlashDen";
- FlexDen[158] – flex specific files including "stock Adobe Flex Creatives, Applications, Components, Air Apps, Libraries, Video Tutorials, Skins, and More"[158];
- Flashloaded[159] – includes a large selection of commercial and free components exclusively for Flash CS* Professional.

These sites are just a snapshot sample of what is available, a quick web search using keywords like "Flash" combined with "components" or "extensions" will uncover many such sites. In all cases it is important to note the license type (e.g., open source), install format (e.g., mxp), and the specific Flash tool (including version) they are designed for (e.g., Flash Professional CS4).

6 Flash Alternatives

The functional breadth of the Flash player has evolved over time into the equivalent of a 'Swiss army knife' of browser plug-ins. As such, it can be difficult to find competing technologies with a similar scope in capabilities for comparison. Many competing web and desktop technologies specialize in what are essentially capability subsets of the Flash platform. For example, if we focus only on media rendering capabilities then we might draw comparisons with Apple's QuickTime player or Microsoft's Windows Media player. If we are interested in 3D web gaming then we would be amiss not to compare Flash with Unity. Website construction would no doubt invoke comparisons with browser specific technologies like HTML and CSS.

Perhaps the most interesting capability area looking forward relates to Flash's potential as an application runtime environment, or more specifically as a platform for desktop and web based RIAs. Unlike the video and web graphics spaces in which Flash dominates, Flash is just one of a few closely competing RIA platforms. Brett Kromkamp provides a good overview[160] on his blog comparing several competing RIA platforms.

6.1 Open Source

Technically speaking, open source alternatives to the Flash player do exist. However, open source players like Gnash, vektrix, and Swfdec have to date lagged the development and feature support of Adobe's player by such a wide margin they are often not viable substitutes. Gnash "supports most SWF v7 features and some SWF v8 and v9"[161] and is really only regularly updated for Linux based systems – "Win32 users can now download an executable based on the 0.8.2 release, which is now seriously out of date, and won't work with YouTube"[161]. As of this writing, Swfdec has not been updated in over a year according to the project's website. In short, as far as Flash runtime clients are concerned, the Adobe owned family of Flash players for desktop, web, and mobile devices are, for most intents and purposes, the only feasible means for playing much of the modern Flash content available on the web today.

OpenLaszlo is an open source tool for building and deploying RIAs to either the Flash player or dynamic HTML (DHTML). "The OpenLaszlo SDK consists of a compiler written in Java, a runtime JavaScript library, and an optional Java servlet that provides additional services to the running application"[162]. OpenLaszlo applications are written in LZX, a declarative style XML language similar to HTML and MXML. The LZX source files can be compiled into executable binaries specifically for the Flash player (currently swf8 and swf9) or browser native DHTML. The optional OpenLaszlo servlet can be deployed to servers in support of applications that require "data integration via SOAP, XML-RPC or Java-RPC, or that require persistent connection capabilities or run-time media transcoding"[162].

In a similar vane as OpenLaszlo, HaXe is an open source Write-Once-Compile-Anywhere (WOCA) platform, consisting mainly of the haXe programming language and haXe compiler. HaXe was created by Nicolas Cannasse to provide developers with "a tool to create websites & applications using a single unified programming language" [163]. Programs written in the haXe programming language can be 'compiled' into native source and byte code versions for

javascript, C++, php, Flash, and Neko (a virtual machine created by Cannasse). Specific Flash targets currently supported are Flash player 6-8 (AVM1), Flash player 9-10 (AVM2), and Flash source files (ActionScript 3). The haXe website contains many haXe specific resources including documentation, tutorials, downloads, community forums, as well as a list of Mac, Windows, and Linux based IDEs, plug-ins and editors which can be used to write and compile haXe programs.

6.2 Microsoft Silverlight

Arguably the closest directly competing technology to Flash in terms of capability breadth is Microsoft's Silverlight. Wikipedia describes Silverlight as "a web application framework that provides functionalities similar to those in Adobe Flash, integrating multimedia, graphics, animations and interactivity into a single runtime environment"[164]. Flash is a comparatively more mature platform, with roots tracing back to the mid 90's and the early days of the internet, while Silverlight is a relatively recent newcomer having been introduced by Microsoft in 2007. The initial Silverlight version 1.0 lacked many capabilities standard with the Flash platform, however in more recent versions Microsoft have been rapidly closing the feature and performance gaps. However, Silverlight still exhibits a certain level of immaturity, having equivalent issues which were encountered and resolved years ago with Flash.

Silverlight does have several notable advantages over Flash, especially with regard to developer tools and communities. Silverlight applications can be developed using any of Microsoft's .NET programming languages in conjunction with its popular Visual Studio and Expression Blend tools. Microsoft also has huge legions of .NET developers which can be tapped for building Silverlight based RIAs. In terms of popularity, Silverlight currently sits at about half of the browser plug-in penetration of Flash. However Microsoft does dominate the desktop OS and browser segments, and will likely continue to leverage this position to further push out the Silverlight runtime to end users.

One of the areas Silverlight currently lacks in comparison to Flash is regarding support and ubiquity across multiple operating systems and devices. While Adobe is aggressively rolling Flash onto every device with a screen, it remains to be seen how well Silverlight is supported on many devices such as smartphones. Note also, development and support of the Linux version of Silverlight (via the open source Moonlight project) significantly lags efforts on Windows platforms.

6.3 JavaFX

JavaFX is Sun's solution for creating RIAs which is tightly integrated with the broader java runtime engine (JRE). JavaFX applications are written using a declarative language called JavaFX Script. JavaFX applications run on any desktop (and browser) with the JRE pre-installed, as well as "easily" integrating with mobile phones running Java Mobile edition (ME). The platform consists of an SDK (e.g., compiler; UI controls; media,graphics, and rich text libraries), the NetBeans IDE, and tools for exporting graphical assets to JavaFX from designer tools such as Adobe Photoshop and Illustrator. JavaFX is a relative RIA platform newcomer having been introduced in 2008, and has yet to make any significant footholds in the RIA segment. It does

however benefit from the comparatively large JRE install base on desktop and mobile systems, as well as the huge java developer community.

6.4 HTML

The World Wide Web Consortium (W3C) is an international organization which develops standards for the web including HTML. HTML, currently at version 4, is the declarative style xml mother language of browsers and the World Wide Web. Generally, web applications which strictly comply with the HTML 4 standard can be rendered properly by all browsers. However, these applications can not include many advanced animation, multimedia, and RIA capabilities, as those features are not defined in the current standard. In this light, Flash can simply be regarded as a complementary technology to HTML 4, rather than a competing one.

HTML extension technologies such as xhtml and AJAX do however add RIA type features, but at the same time introduce incompatibilities across different browser vendors and versions. AJAX technologies also do not really provide any animation and video capabilities. Finally, AJAX based RIAs also have significant obstacles with regard to scalability, as the client server data connections tend to be routed through a single web server.

The upcoming version 5 of HTML does promise to deliver many of the advanced graphics, interactivity, and RIA type capabilities currently available through plug-ins like Silverlight and Flash, but has its own significant challenges. The issues facing HTML 5 are the subject of much ongoing controversy and debate, and relate largely to achieving agreement among the major browser vendors. Section 9 discusses the future of the Flash platform as well as more details surrounding HTML 5.

6.5 Curl

Curl is a relatively small, but noteworthy, Flash alternative for building and deploying RIAs. Curl caters mainly to the business niche of applications within the RIA market, and seems particularly popular within Japan. The Curl programming language was designed for creating interactive web applications and combines elements of text markup, scripting, and object oriented programming. The Curl platform includes the Curl language, runtime engine (RTE), IDE, and several libraries for connecting Curl applications to back-end data servers.

7 Criticisms

Flash is a often a very polarized topic among developers and end users alike; some people swear by it, while others loathe the very mention of the word Flash. The latter group's animosity is reflected by the popularity of browser plug-ins such as FlashBlock which exist solely to enable web surfers to disable Flash content. Some of the criticisms are no doubt justified, some may be out of date, while other complaints are misdirected.

Flash is a tool, and like any tool it can be misused, overused, or simply used incorrectly. For example, the gross overuse of banner advertising and other Flash based content on many websites is a familiar and common complaint. Gaudy advertising aside, there have been too many examples of websites utilizing flashy graphics and animations for simple navigation tasks where HTML based menus would be far faster and effective. Critics will conclude that Flash is not good for website design, however depending on the services and content offered by the web site, Flash may be the best choice for specific elements or components within. Well designed sites use the tools and technologies which match best the purpose or task. An example would be a site that uses HTML based navigation to a page which offers video using a Flash player element within the page. Similarly, a casual gaming site can be laid out in HTML and offer the Flash games as embedded elements therein. However, unlike websites, web applications can often be done entirely in Flash very effectively.

Using Flash design oriented tools such as Flash Professional, it is all too easy for beginners and non-developers to immediately jump-in and make 'something'. The problem is these makeshift creations can easily become runaway performance hogs, crashing browsers or crippling the performance of even high-end systems.

Arguably the biggest longstanding gripe concerning the Flash platform relates to the simple fact it is largely a proprietary solution. Adobe have made some progress in this area by open sourcing portions of the platform such as the Flash player's virtual machine (code named Tamarin), the text layout and Flex frameworks, as well as the BlazeDS data server. While there are arguments both for and against open source software, the general consensus (and past experiences) seems to favour a free and open development model. The notion of any particular company or entity having essentially monopolistic control over any significant piece of technological infrastructure does not usually bode well for many stakeholders whether they be end-users, developers, governments, or industry as a whole.

Forcing the page-centric paradigm of the browser to act as an application runtime environment often manifests as difficulties related to both interaction and search engine optimization (SEO) or indexing. From an interactivity perspective, users complain Flash breaks the standard browser interaction model. For example, Flash content may cause a loss of browser focus such that keyboard shortcuts, the mouse's scroll wheel, and the context menu do not function as expected until the user clicks outside the Flash content area. Even more frustrating is the loss of basic browser navigation functions (e.g., back and forward buttons) and browsing history. Many of these SEO and browser interaction problems can be resolved using a popular open source tool known as SWFAddress. This small library allows developers to "create unique virtual URLs that can point to a website section or an application state"[165]. Adobe have also been working with

major search engine providers (notably Google and Yahoo) to address the SEO challenges and there is a large section on the Adobe developers website dedicated to dealing with SEO.

Mac and Linux users claim Flash player support is heavily skewed towards Windows 32-bit based systems[166]. They point out the relatively poor performance of the Flash player on non-Windows platforms as well as longstanding bugs that persist on Mac and Linux versions of the Flash player. Adobe has cited Apple's lack of co-operation as a contributing factor to Flash performance on the Mac. Another parallel complaint is the lack of 64-bit versions of the Flash player. To date Adobe has released a beta 64-bit version of the Flash player for Linux only. The Flash player's lack of support for multi-threading is also a popular source for criticism. The pixel bender API (Section 2.2.2) does however offer a partial solution, as it supports multi-threading.

The current lack of Flash player on Apple's iPhone OS has been a source of both frustration and confusion for both developers and end-users. This notorious and contentious issue is discussed in Section 2.7. Given Apple's current official stance, the Flash runtime is not likely to be permitted on the iPhone anytime soon. Adobe's recent tooling support for compiling Flash code into native iPhone applications does offer a technically feasible WOCA style solution for Flash developers.

There has been an assortment of complaints relating to privacy when using the Flash player. Most of these concerns relate to the use of local shared objects (LSOs) which are Flash's equivalent to browser 'cookies'. Like cookies, LSOs are used by websites to track and store information about how visitors are using their website. Unlike cookies however, LSO's are not as well known by the public, and are typically oblivious to the user's privacy settings in their web browser.

In some ways Flash has been a victim of its own popularity and rapid evolution as a platform. A case in point is the abundance of outdated tutorials, code samples, and other developer resources which persist online. While successive versions and updates of the Flash player itself offer new capabilities, they can simultaneously completely change programming tactics. For example Flash player's recent update from version 10 to 10.1 introduced an extensive list of new capabilities. While the new features are most welcome, they can nonetheless significantly change or deprecate many existing (relatively recent) coding APIs and techniques.

Perhaps not surprising for any prevalent platform associated with the internet, there continues to be a steady flow of security concerns associated with Flash[167]. Adobe maintains a growing list of official security advisories and patches for all their software on the security support section of their website.

Adobe has found themselves in a catch-22 situation with regards to many Flash player bugs and issues. Often fixing these problems can 'break' many existing applications which were coded around (or even depend on) the defective behaviour. Historically Adobe have adopted a 'don't break the web' approach to such bugs, more specifically opting to allow the bug to persist indefinitely unless the application is republished for a newer version of the Flash player which has rectified the defect. The Flash player will check the specific version of the swf content before playing such that a bug in version 9 of the player will persist in version 10 of the player when playing content published for version 9 (swf 9).

Some negative feedback has been targeted to specific Flash tools and their release versions. Such is the case with the (original) production release of Flash Professional CS4. Adobe did eventually address most of the bugs in a subsequent update, but the extensive list of fixes does enforce the argument the original release was premature.

Video content providers have long complained of the lack of a proper digital rights management (DRM) solution for the Flash platform. Adobe has recently responded to these concerns with the release of Flash Access 2 – Adobe's content protection solution.

Finally, developers will note the ActionScript 3 language does not offer certain elements available in other modern OOP languages including "private constructors, generics, enums, and method overloading"[160].

8 Resource Links

This section provides a selection of learning, development, community, and other resources specific to the Flash platform which the author has found useful. The information provided represents but a small sample of the (seemingly) endless resources available online and is by no means exhaustive. The intent is to offer a few good launch points for exploring the Flash world.

Also note access to some resources such as developer communities and digital publications may be restricted or limited without a (free) membership.

8.1 General

Adobe Developer Connection – The developer area of Adobe's website is extensive and is further subdivided by product categories (e.g., AIR, Flash Professional, Flex) or technologies (e.g., ActionScript, swf, video). The site is comprehensive and provides an excellent starting place for developers new to the Flash platform with tutorials, online forums, sample code, demos, downloads, documentation, articles, and much more.

Cheat Sheets – The authors of the Web Design Tools website provide a compilation of Flash and ActionScript oriented 'cheat sheets' - concise and printable quick reference guides.

DZone Refcardz – A collection of professionally made developer 'cheat sheets', each dedicated to a specific Flash development tool, language, or technology.

Flash Enabled – Templates, galleries, news, reviews, tutorials, links, and other resources related to Flash, Flex, and AIR.

Flash & Flex Developer's Magazine – A free digital (pdf) magazine published bi-monthly and dedicated to all things related to developing applications for the Flash platform.

HexoSearch – A search engine dedicated to ActionScript and Flex with results sorted by community voting. There is also a plug-in version for the Firefox web browser.

InsideRIA - An online news and blog aggregation site sponsored by Adobe and dedicated to the topic of rich internet applications.

Stack Overflow – A popular question and answer site (similar to Yahoo Answers) for developers regardless of the programming platform. Uses tagging and peer voting to aid searches.

8.2 Learning

Adobe TV - Great source of online training videos for all Adobe products including Flash Professional, Flash Builder, and Flash Catalyst. Browse by product or subject category. Difficulty levels range from absolute beginner through intermediate to advanced.

Adobe AIR Tutorials for Web Developers – A compilation of 35 training tutorials for developing and deploying desktop (AIR) applications provided by web and graphic design company Vandelay Design.

Flash and Math – A well organized web site for learning ActionScript programming and featuring numerous tutorials organized by difficulty (e.g., beginner, intermediate, and advanced).

FlashPerfection - Huge collection of free Flash tutorials organized by numerous categories including sub-sections for animation, math and physics, audio, interactivity, video, back-end, basic, 3D, and many more.

gotoandlearn – Highly recommended Flash learning web site; contains a variety of free online video tutorials using Flash, Flex, and AIR by Adobe evangelist Lee Brimelow.

How Do I Learn Flex – Sean Moore's ('SeanTheFlexGuy') excellent learning guide provides an organized collection of learning links and paths.

Learn Flex 4 from scratch – Adobe evangelist Ted Patrick provides a short series of Flex 4 introduction tutorials for developers.

Lynda.com – Online learning site featuring hundreds of professional video training courses (both for purchase and free) including many Adobe software products like Flash Professional, Photoshop, Illustrator, AIR, and Flex Builder.

The tech labs – A learning and "information gateway" site for Flash, Flex, and AIR with tutorials, educational articles, tips, reviews, and news.

10 Animation Tutorials – A great collection of 10 simple step-by-step Flash animation tutorials for beginners.

25+ Very Useful Flash & ActionScript 3 Tutorials – A recent collection of beginner and intermediate Flash learning tutorials. Each listing entry includes a brief overview, demo or snapshot, and a link to the actual tutorial.

8.3 Communities

Actionscript.org – A website dedicated to ActionScript developers featuring hundreds of Flash and Flex tutorials, articles, and a large online forum.

Adobe Forums – Adobe's official online support forums for end users organized by Adobe product.

Adobe Groups – Hundreds of user groups from around the world organized by geography (e.g., continent and city) as well as Adobe product.

CFLEX: Community Flex – A very large site dedicated to developing rich internet applications using Flex. "The goal of the site is to aggregate tips, lessons, news, and articles from around the net."

Flash Kit – A large online community of over half a million Flash developers. The site includes hundreds of tutorials and several large (downloadable) collections of Flash 'raw materials' (e.g., sound, image, and graphics files).

FlexCoders – A Yahoo! Groups community (currently 12,000+) specializing in the development of rich internet applications using Flex, MXML and ActionScript 3. Topics also comprise "best-practices for testing, design patterns, J2EE and .NET integration, etc."

8.4 Blogs

Adobe Blogs – A large directory of Adobe (mostly employee) blogs. Each listing has a brief description of the blog, a URL link to the blog, and the number of blog entries and comments.

Adobe AIR Blog – Adobe's official AIR blog provides reviews, announcements, and other information specific to the AIR community.

Adobe Flash Platform Blog – Adobe's official Flash platform blog aggregates information from many Adobe bloggers to "provide the latest news, updates, and insights into the technologies, tools, and partners across the Flash Platform".

Adobe Flex Blog – Adobe's official Flex blog contains important news, contests, and announcements specific to the Flex developer community.

EverythingFlex - Rich Tretola is a veteran Flex developer and author. His blog is a great source for news, tutorials, and demos related to Flex and AIR development.

Flex Examples - Peter deHaan of the Adobe Flex QA team maintains a blog containing several well written examples in Flex development.

Quietly Scheming - Ely Greenfield's blog provides some excellent code samples and Flex components, particularly for charting.

RIA Cowboy – Adobe Flex evangelist James Ward "primarily uses Flex to build beautiful front-ends for Java based back-ends". His blog contains interesting and informative posts related to Flex, rich internet applications, and pdf.

swfGeek - Dave Gamez is a veteran Flash developer who uses his blog to aggregate Flash related news, articles, and links.

Ted Patrick – Ted Patrick is a senior Adobe spokesman who often uses his blog to showcase the latest and greatest capabilities of the Flash platform.

TheFlashBlog – Developer and Adobe evangelist Lee Brimelow maintains a Flash platform blog with lots of news, videos, demos, announcements, contests, and links to other Flash resources.

8.5 Events

Flash Platform Event Calendar – Adobe's Google calendar of Flash Platform events from around the world. Note these are all events Adobe is sponsoring or participating in.

Adobe Max – Adobe's largest annual conference draws thousands of designers and developers and features numerous activities including hands-on labs, training sessions, community meetings, keynote presentations, as well as new product announcements and demonstrations from Adobe and its industry partners.

FITC – 'Flash in the Can' originated as a festival for Flash designers, artists, and developers and has grown into several events each year hosted around the globe. FITC events are typically focussed on Flash design and animation topics, however other Flash themes such as mobile, AIR, and Flex development have increasingly been appearing.

Flash on the Beach – The popular annual UK Flash conference typically covers three tracks – design, technical, and inspirational as well as Flex and AIR sessions.

Web Conference Round-up - Smashing Magazine's "comprehensive list of web development and graphic design-related conferences and events" from around the globe. The listing is subdivided into conference categories such as general web design, back-end programming, AJAX, User Interface, and Flash. Each listed conference contains a brief overview and location information.

360|FLEX – A community-driven event for Flex developers held at multiple times and locations around the globe. The aim of these conferences is to "bring the best of the Flex community together in one place to share war stories from the trenches and to allow the experts to share their deep technical knowledge with the community at large."

8.6 APIs, Libraries, Frameworks

8.6.1 API Collections

ActionScript 3.0 Reference for the Adobe Flash Platform – Adobe's official API for the entire Flash platform including the Flash player, AIR, Flex, and LiveCycle DS. "Contains the ActionScript language elements, core libraries, and component packages and classes for the tools, runtimes, services and servers in the Flash Platform."

Adobe Cookbooks – Search and/or share code 'recipes' – code snippets for common programming tasks. The listings can be sorted and filtered by Adobe development tool categories.

AS3 Code Libraries – Adrian Parr has put together an extensive list of Flash APIs and organized them by categories like '3D Physics Engines', 'Augmented Reality', 'Security', 'Particle Systems', 'OOP Frameworks', and many more.

Data Visualization Libraries – A compiled list of 28+ rich data visualization tools complete with descriptions, previews, and links. Compiled by Theresa Neil at InsideRIA and includes Flash, Flex, AJAX, and Silverlight libraries.

Flex Frameworks – Moxie Zhang of InfoQ has put together an excellent collection of Flex RIA building tools. The list includes unit testing libraries, back-end tools, IDEs, and numerous Flex design pattern frameworks like Mate, Swiz, Cairngorm, PureMVC, and Parsely.

Flex3 UI Controls and Style Explorer – This is a quick and handy Flex based web application which allows you to view and interact with many of the out-of-box UI controls and components available in Flex Builder 3.

Nine Animation Engines – Overview and links to nine different third-party animation tools for Flash.

Open Source Flash – Home of everything Flash and open source, including tools, frameworks, IDEs, players, APIs, and several other open source Flash tools.

Physics Engines – A small compilation of Flash 2D and 3D physics engines complete with brief descriptions from Emanuele Feronato's blog.

SeanTheFlexGuy – Sean Moore is a veteran and certified Flash/Flex developer who routinely uses his blog to post compilations of useful tools, libraries, and APIs specific to Flash and Flex development. Examples include 'List of 22 ActionScript 3.0 API's', '36 New, Cool Flex and AS3 Tools, Libraries and Components', and 'List of 34 More ActionScript 3.0 APIs'.

Spark Project – A community dedicated to open source and ActionScript development. The site hosts an extensive collection of links to Flash specific tools and APIs grouped by categories such as graphics, 3D, math, sound, network, and text.

3D Engines – Ding ZhiGang's blog (ntt.cc) provides a round-up of several popular Flash 3D engines such as Papervsion3D, FIVe3D, Sandy 3D, Alternativa3D, and CopperCube.

40 ActionScript 3 Cloud/Service API's – Ted Patrick's list of cloud based APIs for Flash developers. Examples include AS3 libraries for accessing services provided by eBay, Fedex, ESRI, facebook, Amazon, Flickr, yahoo, and youtube.

104 Free Open Source APIs, Libraries, and tools for the Flash Platform – Exactly as the title implies of this entry on Vipin Chandran's blog (The Flashchemist). Includes links for several non-Google hosted libraries, unit-testing frameworks, and AMF remoting implementations.

8.6.2 Miscellaneous APIs

AlivePDF - An open-source ActionScript 3 (Flash, Flex, AIR) library for generating PDF documents from within a Flash application.

FlexPaper - An open-source Flex (client-side) component created by Erik Devaldi which enables viewing, searching, and printing pdf documents from within the Flash player.

Google Maps – Google Maps's Flash API hosted on Google code. Note a Google Maps API key is required for using the Google mapping service.

Google Wave – ActionScript client library for Google Wave hosted on Google code. The wiki includes getting started tutorials, sample projects, and code snippets.

Merapi – An open source framework that enables direct communication between Flash (AIR) and java applications both running on a user's desktop (client-side).

NeuroSky MindSet BCI – An open source ActionScript 3 API (created by Sean Moore) for developing Flash applications which harness input from NeuroSky's MindSet; a brain-computer interface headset which senses human brainwave activity.

UMap – A universal mapping API for ActionScript 3 which enables developers to easily create rich interactive maps using many map service providers including OpenStreetMap, Bing Maps, CloudMade, Yahoo, and ArcGIS.

WiiFlash – An open source API for controlling Flash applications using Nintendo's Wii remote ('Wiimote').

8.7 Components

Refer to Section 5.1.2 for information and links to several online sources for Flash components.

9 Future

Adobe has made several recent announcements regarding their near future (read 2010) plans for the Flash platform. In his keynote address at Adobe Max 2009[168], CEO Shantanu Narayen summarized the main technological trends his company sees looking forward into 2010 and beyond. In general terms, Shantanu expects a continued rapid growth in digital devices with increasingly 'significant' capabilities, particularly within the mobile space. He also predicts the lines to continue to blur between content and application, in addition to applications generally becoming more social and collaborative. From a developer perspective, the Adobe CEO expects the explosive growth of online content and services to carry on, with more and more leveraging of service-oriented architectures.

On the client side, Adobe is aggressively establishing the Flash player on mobile phones, tablets, set-top boxes, and pretty much any digital device with a screen. Given the significant level of buy-in for the Adobe led Open Screen Project by tech industry leaders, it is quite possible Flash penetration on these devices could eventually reach the same ubiquitous levels Flash enjoys on PCs. By formally partnering with 19 of the world's top 20 smartphone manufacturers, Adobe have assured a dominant Flash presence across this rapidly growing device market. The lone iPhone holdout will likely continue to be a prominent gap in an otherwise complete WORA sweep of the mobile market for the Flash platform. Adobe have attempted to work around Apple by adding the ability to compile Flash projects into native iPhone apps.

Figure 99: Smartphones are a big part of Flash's future [169].

The significance of the handheld market cannot be overstated; the sheer number of these increasingly capable computing platforms is quickly surpassing desktop PCs[62-63, 170]. Each new smartphone generation continues to erode away many PC duties while adding new mobile specific capabilities such as location awareness and augmented reality functions.

The last frontier in the Flash quest for global client-side dominance is what Adobe refers to as the 'third screen' - digital living room devices[65] such as digital TVs, DVD players and advanced set-top boxes with internet connectivity.

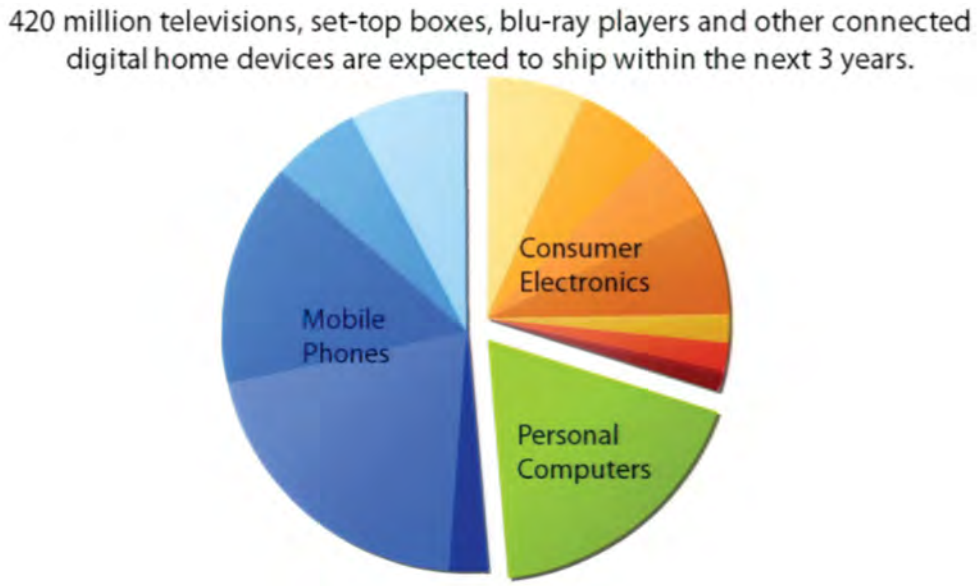

Figure 100: The traditional PC in its role as an information gateway is increasingly overshadowed by mobile phones and the 'third screen' - digital appliances. (image from [1])

In part, Adobe are wagering there will be a shift away from traditional media broadcasting towards internet sources like Hulu, Last.fm, netflix, and YouTube. Adobe are not alone in this prediction, and have for example been working with companies like Intel to deliver Flash enabled digital living room devices[171].

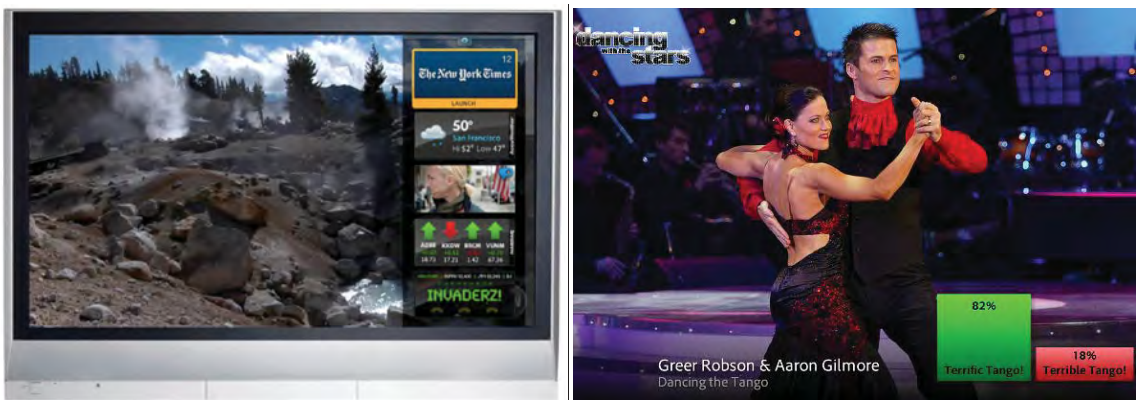

Figure 101: Televisions and set-top boxes that come standard with the Flash player will enable viewers to interact and even participate in the content they watch (image from [1]).

The primary use cases include hi-definition internet based video, information widgets, notifications, and social/audience interactivity (Figure 101). Whether or not this latest effort towards convergence between PCs and entertainment appliances will thrive remains to be seen.

Adobe has also indicated a desire to re-unify the Flash runtime to a common core build on all devices. More specifically this means focussing on optimizing the full version Flash player (10.1), and slowly moving away from the less capable and lighter weight Flash Lite player. This trend will no doubt be aided by the rapidly advancing hardware capabilities of 'low-end' mobile devices.

Adobe is pushing the AIR platform as a best of both worlds (online and offline) application platform. AIR based digital publications like the NY Times Reader 2.0[38] and the soon to be released Sports Illustrated Digital Magazine are promising examples which change the way users consume information.

"The tablet format is much easier on the eyes than reading the same story on the Web, and you get the added bonus of full-screen slide shows or videos. You can also flip through photos within the text, while continuing to read. Sports scores and other data can be dynamically updated from the Web, or you can share stories and photos via email, Facebook, or Twitter."[172]

Figure 102: A yet to be announced tablet edition of Sports Illustrated Digital Magazine[172] powered by Adobe AIR.

The competition within the RIA market for a next generation web application platform is very tight with competing solutions such as javaFX, Silverlight, Flash, ajax, and html 5. Flash's steadily evolving technologies and tools for building and deploying RIAs combined with its growing ubiquity seem likely to only strengthen it frontrunner position as a RIA platform.

On the surface and looking forward, html 5 in particular would seem poised as a heavily favoured platform with the greatest potential for ushering in the next generation of RIAs. Re-defining the html standard and related built-in browser technologies to support sophisticated user interfaces including advanced data binding and user controls, rich graphics and animation effects, multimedia, and diverse interactivity would indeed seem a very logical next step in the progression of web browsers. Indeed many of the upcoming features of html 5 add support for many RIA like capabilities. Examples include the new 'canvas' tag for JavaScript powered graphics, the video element for native browser video support, geo-location APIs for enabling location aware web apps, application caches for offline browser functionality, web sockets for persistent connections, and 'web workers' which enable background processing of specific web app tasks. It is interesting to note that these, and in general all of the proposed new built-in browser features, are a subset of capabilities already available using existing browser runtime solutions such as Silverlight and Flash.

Many of the upcoming RIA-like features of html 5 face a number of significant technical and political obstacles which will likely impede or slow its adoption. The reasons behind the lack of progress and support for html 5 are the subject of much ongoing disappointment, criticism, and

debate [173-176]. Many of the issues trace back to the fact the browser was never intended or designed as an application runtime environment, and attempts to make it so often involve a kludgy mishmash of workaround technologies. Unfortunately many of the new RIA-like features proposed in html 5 are not universally supported by the matrix of web browsers and their respective versions (see Table 6). A fundamental issue with all browser innovations is simply the very long uptake for people to upgrade their browsers. In order to get the penetration numbers seen in Flash, you generally have to wait 5 years once the innovations are in all the major browsers.

Table 6: Past, Present, and Future Browser Compatibility with HTML 5[173].

	Internet Explorer	Firefox	Safari	Chrome	Opera
Far Past	6.0: 4%	2.0: 35%	3.1: 43%	1.0: 54%	9.0: 34%
Past	7.0: 12%	3.0: 48%	3.2: 66%	2.0: 77%	9.6: 55%
Present	8.0: 28%	3.5: 79%	4.0: 87%	3.0: 86%	10.0: 60%
Near Future (2009)	8.0: 28%	3.6: 81%	4.*: 89%	3.0: 86%	10.0: 60%
Future (2010 or later)	9.0: 32%	4.0: 87%	4.*: 89%	4.0: 88%	10.*: 71%

Microsoft's Internet Explorer's historically lacking support for newer RIA enabling web technologies is a particular pinch point, especially given its majority share of the web browser market (see Figure 65). Essentially what all this boils down to for web app developers is a painful trade-off between native browser compatibility and RIA like functionality. In other words it is possible to develop RIAs that feature sophisticated user interfaces using built-in browser functionality (e.g., html 5), but most people will not have a web browser able to run it. Suffice to say, while RIA technologies are currently available using browser plug-in architectures like Flash and Silverlight, do not expect to see equivalent features delivered using html 5 as commonplace anytime soon.

10 Summary

The Flash platform is a comprehensive infrastructure of runtime clients, developer tools, and server-side technologies for designing, developing and delivering an extensive variety of software content and applications. The Flash player is unequivocally the most ubiquitous runtime client in existence, having a global presence that proliferates across competing browsers, desktops, netbooks, handhelds, and countless other existing and emerging digital devices. By incorporating the Flash player into the pdf platform, Adobe have further expanded Flash's reach, while redefining and blurring the boundaries between document and application.

The broad diversity of Flash content reflects the core strengths of the platform including audio, video, graphics, images, animations, advanced text, and sophisticated interactivity. Arguably more impressive is the platform's capabilities related to developing and deploying rich internet applications (RIAs). More specifically this includes Flex, the RIA building framework for the Flash platform. Flex includes server-side connection, remoting, and data transfer technologies; expansive libraries of open source and third-party components and APIs; and powerful IDEs like Flash Builder which enable developers to effectively leverage all these tools. Examples of Flash applications include websites, games, advanced visualizations, videos, digital publications, and an increasing number of data driven RIAs. These second generation web applications span multiple industry, academic, and government domains and increasingly incorporate social networking, GIS, collaboration, and many other cloud based services. New breeds of web mash-up applications that can leverage the rapidly growing number of services and resources available across the internet can now be readily developed using RIA based solutions such as Flex.

The largely proprietary nature of Flash, as well as the relatively poor performance of many Flash applications (particularly on the Mac OS), are two common and persistent criticisms of the platform. No doubt due to the successful penetration levels of the Flash player, it is increasingly the target of hackers and security is emerging as another significant concern for the platform.

As a research tool Flash offers much potential for design, rapid prototyping, development, and experimentation across a plethora of devices, utilizing proven architectures and tools for optimized visualizations, rich internet applications, rich interactivity, data connectivity and remoting technologies. Flash based applications provide a tangible means for exploring, refining, and implementing interface design concepts as well as offering a mechanism for soliciting feedback from SMEs. Adobe offers a plethora of powerful and extensible Flash tools for rapid application development and content creation. In addition, numerous Flash based developer tools such as IDEs, components, frameworks, and APIs are available from a myriad of third-party and open source communities. Applications can be built using out-of-box components, customized components, third-party components, or built entirely from scratch.

Historically Flash has been securely established as 'the' platform for graphics and animation on the web. More recently, Flash has achieved critical mass in the online video space with over 80% of all internet video being Flash based[1]. Looking forward, Flash is an early frontrunner in the battle for RIA platforms, thanks in large part to its impressive levels of ubiquity and RIA enabling technologies like Flex.

References

[1] Balakrishnan, M. (5 Oct 2009), Adobe Max 2009: Flash Platform Runtimes - What is Coming? (online), Adobe, http://tv.adobe.com/watch/max-2009-envision/roadmap-flash-platform-runtimes/ (Access date: 28 Oct 2009).

[2] Lynch, K. (5 Oct 2009), Adobe Max 2009: Keynote Day 1 (online), Adobe, http://tv.adobe.com/watch/max-2009-envision/max-2009-keynote-day-1/ (Access date: 29 Oct 2009).

[3] Adobe.com Flash Player Penetration Statistics (online), http://www.adobe.com/products/player_census/flashplayer/version_penetration.html (Access date: 10 July 2009).

[4] adobe.com Adobe Flash Platform: Create and deliver rich applications and video (online), http://www.adobe.com/devnet/flashplatform/ (Access date: 26 Nov 2009).

[5] adobe.com Flash Player 10 Features: Double Identity (online), http://www.adobe.com/products/flashplayer/features/ (Access date: 26 Sep 2009).

[6] Adobe.com Tour De Flex (online), http://www.adobe.com/devnet/flex/tourdeflex/ (Access date: 19 Oct 2009).

[7] adobe.com Media sample: Image masking (online), http://www.adobe.com/devnet/flash/samples/media_3/index.html (Access date: 13 Jul 2009).

[8] adobe.com Drawing sample: Drawing API (online), http://www.adobe.com/devnet/flash/samples/drawing_3/index.html (Access date: 28 Jul 2008).

[9] Tynjala, J. Logicly — A logic gate simulation (online), http://joshblog.net/projects/logic-gate-simulator/ (Access date: 23 Apr 2008).

[10] Peters, K. BIT-101 Blog: Flash 10 3D vs. “The Old Fashioned Way” (online), http://www.bit-101.com/blog/?p=1674 (Access date: 24 Sep 2009).

[11] flashpanoramas.com Flash Panorama Player (online), http://flashpanoramas.com/player/ (Access date: 4 Dec 2009).

[12] Google Google Maps Street View (online), http://maps.google.ca/intl/en_ca/help/maps/streetview/ (Access date: 4 Dec 2009).

[13] adobe.com Pixel Bender (online), http://labs.adobe.com/technologies/pixelbender/ (Access date: 1 Dec 2009).

[14] adobe.com Pixel Bender Exchange (online), http://www.adobe.ca/cfusion/exchange/index.cfm?s=3&o=desc&exc=26&event=productHome&from=1 (Access date: 1 Dec 2009).

[15] Kupila, A. Pixel Bender Levels Example (online), http://www.anttikupila.com/flash/pixel-bender-levels-example/ (Access date: 24 oct 2009).

[16] Elrom, E. Using Pixel Bender to calculate information (online), http://www.flashmagazine.com/tutorials/detail/using_pixel_bender_to_calculate_information/ (Access date: 1 Dec 2009).

[17] adobe.com Producing audio for the web using Soundbooth and Flash (online), http://www.adobe.com/devnet/flash/articles/web_audio_02.html (Access date: 4 Dec 2009).

[18] Sonoflash.com Sonoflash (online), http://sonoflash.com/Home.htm (Access date: 24 oct 2009).

[19] Adobe.com Tenoran (online), http://www.adobe.com/jp/devnet/flash/articles/flp10_sound_02.html (Access date: 24 Oct 2009).

[20] noteflight.com noteflight (online), http://www.noteflight.com/scores/view/ca3a1593ed6236c9bb56971b89d80f23a105f5c4 (Access date: 24 Oct 2009).

[21] Hobnox Audiotool (online), http://www.hobnox.com/audiotool (Access.

[22] aviary.com Myna Audio Editor (online), http://aviary.com/tools/myna (Access date: 8 Sept 2008).

[23] Adobe.com Flash Player 10 features (online), http://www.adobe.com/products/flashplayer/features/ (Access date: 23 oct 2009).

[24] immersivemedia.com/ Immersive Media (online), http://www.immersivemedia.com (Access date: 26 Oct 2009).

[25] invism.com RealityV (online), http://www.invism.com/technology/reality-v/reality-v.html (Access date: 10 Nov 2009).

[26] Adobe.com Advanced Technology Labs (online), http://www.adobe.com/technology/ (Access date: 26 Oct 2009).

[27] Goldman, D.B., Gonterman, C., Curless, B., Salesin, D., and Seitz, S. M. (2008), Video object annotation, navigation, and composition, From the UIST '08: Proceedings of the 21st annual ACM symposuim on User Interface Software and Technology.

[28] Casperson, M. Control a Flex Video Application Using Mouse Gestures (online), http://active.tutsplus.com/tutorials/flex/control-a-flex-video-application-using-mouse-gestures/ (Access date: 4 Dec 2009).

[29] Coady, D. and McMullin, C. (2009), Exploring Operator Interface Design Concepts Using Adobe Flash, (DRDC Atlantic TM 2009-007).

[30] Lin, M. Flex+AS3=>Genie Effect (online), http://masolin.blogspot.com/2008/10/flexas3genie-effect.html (Access date: 4 Dec 2009).

[31] gapminder.org Gapminder Foundation (online), http://www.gapminder.org/ (Access date: 30 Oct 2009).

[32] University_of_Utah Gentic Science Learning Center: CELL SIZE AND SCALE (online), http://learn.genetics.utah.edu/content/begin/cells/scale/ (Access date: 2 Oct 2009).

[33] GreenSock Getting Started Tweening (online), http://blog.greensock.com/get-started-tweening/#plugins (Access date: 2 Nov 2009).

[34] Brimelow, L. Create a Simple Inverse Kinematics Animation with Flash CS4 (online), http://www.flashperfection.com/tutorials/Create-a-Simple-Inverse-Kinematics-Animation-with-Flash-CS4-54557.html (Access date: 4 Dec 2009).

[35] Adobe_Labs Text Layout Framework (online), http://labs.adobe.com/technologies/textlayout/ (Access date: 21 Oct 2009).

[36] adobe.com Text Layout Framework: Beta 1 Release Notes (online), http://labs.adobe.com/technologies/textlayout/releasenotes.html (Access date: 28 Oct 2009).

[37] adobe.com Drawing Text (online), http://download.macromedia.com/pub/flashplayer/demos/textOnPath/textOnPath.html (Access date: 24 oct 2009).

[38] Larson, R. The New York Times: Sneak Peek of Times Reader 2.0 (online), http://firstlook.blogs.nytimes.com/2009/05/08/sneak-peek-of-times-reader-20/ (Access date: 11 May 2009).

[39] adobe.com About text in Flash Lite (online), http://livedocs.adobe.com/flash/9.0/main/wwhelp/wwhimpl/common/html/wwhelp.htm?context=LiveDocs_Parts&file=00004676.html (Access date: 4 Dec 2009).

[40] crosby, j. Flex: Simple Predictive Text Example (online), http://office.realeyesmedia.com/blogs/john/index.php/2007/03/19/flex-simple-predictive-text-example/ (Access date: 4 Nov 2008).

[41] powercursor.com PowerCursor (online), http://www.powercursor.com/ (Access date: 11 Mar 2008).

[42] Brun, D. Mouse Gesture Recognition (online), http://www.bytearray.org/?p=91 (Access date: 18 Jan 2010).

[43] adobe.com Flex Framework (online), http://www.adobe.com/products/flex/features/flex_framework/ (Access date: 13 Nov 2009).

[44] Wroblewski, L. AJAX & Interface Design (online), http://www.lukew.com/resources/articles/ajax_design.asp (Access date: 5 Dec 2009).

[45] Morris, S. A Rose By Any Other Name (online), http://weblogs.java.net/blog/javakiddy/archive/2007/06/a_rose_by_any_o.html (Access date: 23 Nov 2009).

[46] searchsoa.techtarget.com Rich Internet Application (RIA) (online), http://searchsoa.techtarget.com/sDefinition/0,,sid26_gci1273937,00.html (Access date: 13 Nov 2009).

[47] virginia.gov Virginia Interoperability Picture for Emergency Response (VIPER) (online), https://cop.vdem.virginia.gov/viper/ (Access date: 10 Dec 2009).

[48] esri.com VIPER: Virginia Deploys Web-Based Emergency Management SystemVirginia Department of Emergency Management (online), http://www.esri.com/news/arcnews/fall09articles/viper.html (Access date: 10 Dec 2009).

[49] Shih, T. Flash Innovation Timeline (online), http://ticore.blogspot.com/2009/11/flash-innovation-timeline.html (Access date: 7 Mar 2011).

[50] adobe.com Adobe Flash Lite (online), http://www.adobe.com/products/flashlite/ (Access date: 27 Nov 2009).

[51] Lynch, K. (2008), Kevin Lynch on Adobe's AIR: Extending the Web beyond the Browser, *Knowledge@Wharton* (electronic journal).

[52] Adobe.com Adobe AIR - Browser vs. Desktop (online), http://www.adobe.com/products/air/comparison/ (Access date: 14 Jul 2009).

[53] adobe.com Showcase Applications (online), http://www.adobe.com/products/air/showcase/ (Access date: 9 Dec 2009).

[54] adobe.com Portable Document Format (PDF) (online), http://www.adobe.com/pdf/about/history/ (Access date: 9 Dec 2009).

[55] adobe.com Security for Flash Player compatible content in Acrobat 9 (online), http://www.adobe.com/devnet/acrobat/pdfs/acrobat_reader9_flash_security.pdf (Access date: 13 Oct 2009).

[56] Ward, J. Portable RIAs – Flex Apps in PDFs (online), http://www.jamesward.com/blog/2008/11/05/portable-rias-flex-apps-in-pdfs/ (Access date: 5 Nov 2009).

[57] Devaldi, E. FlexPaper (online), http://www.devaldi.com/?page_id=260 (Access date: 21 Dec 2009).

[58] Adobe.com Adobe Flash Player Versions (online), http://www.adobe.com/software/flash/about/ (Access date: 21 July 2009).

[59] Ward, J. Portable RIAs – Flex Apps in PDFs (online), http://www.jamesward.com/2009/09/13/rias-on-the-web-on-the-desktop-and-in-a-pdf/ (Access date: 5 Nov 2008).

[60] Polanco, J. Flash Player Internals 10.1 Recap (part one) (online), http://www.developmentarc.com/site/2009/10/flash-player-internals-101-recap-part-one/ (Access date: 9 Dec 2009).

[61] Burke, J. Adobe points finger at Apple over Flash for iPhone (online), http://www.tuaw.com/2009/11/03/adobe-points-finger-at-apple-over-flash-for-iphone/ (Access date: 3 Nov 2009).

[62] gartner.com Not surprising: iPhone doubles global smartphone market share! (online), http://www.intomobile.com/2009/05/21/not-surprising-iphone-doubles-global-smartphone-market-share.html (Access date: 9 Nov 2009).

[63] gartner.com Gartner Says PC Vendors Eyeing Booming Smartphone Market (online), http://www.gartner.com/it/page.jsp?id=1215932 (Access date: 9 Nov 2009).

[64] Snol, L. More Smartphones Than Desktop PCs by 2011 (online), http://www.pcworld.com/article/171380/ (Access date: 11 Dec 2009).

[65] Tweney, D. Adobe Flash for Your TV Means Hulu in Your Living Room (online), http://www.wired.com/gadgetlab/2009/04/adobe-flash-for/ (Access date: 20 Apr 2009).

[66] openscreenproject.org Open Screen Project (online), http://www.openscreenproject.org/ (Access date: 5 Nov 2009).

[67] Williams, I. IDF 2009: Adobe demos Flash Player 10 (online), http://www.theinquirer.net/inquirer/news/1556374/adobe-demos-flash-player (Access date: 25 Sep 2009).

[68] finetune.com finetune (online), http://www.finetune.com/FinetuneFamily/ (Access date: 10 Jun 2009).

[69] Wikipedia (29 November 2009), Adobe Flash Media Server (online), Wikipedia, The Free Encyclopedia., http://en.wikipedia.org/wiki/Adobe_Flash_Media_Server (Access date: 10 dec 2009).

[70] osflash.org Red5 : Open Source Flash Server (online), http://osflash.org/red5 (Access date: 9 Dec 2009).

[71] wowzamedia.com Wowza Media Server Pro: A Fully Interactive Flash Media Server (online), http://www.wowzamedia.com/index.html (Access date: 9 Dec 2009).

[72] AskMeFlash.com Comparison Wowza vs FMS vs Red5 (online), http://askmeflash.com/article/10/comparison-wowza-vs-fms-vs-red5 (Access date: 9 Dec 2009).

[73] Smith, J. 1,000,000 Status Updates on CNN Live with Facebook During Obama Inauguration (online), Inside Facebook, http://www.insidefacebook.com/2009/01/20/600000-status-updates-on-cnn-live-with-facebook-during-obama-inauguration/ (Access date: 20 Jan 2009).

[74] ustream.tv Ustream.TV (online), http://www.ustream.tv/about (Access date: 30 Oct 2009).

[75] Knight, R. Blaze Data Services or LiveCycle Data Services? (online), InfoQ, http://www.infoq.com/articles/Blaze-LiveCycle (Access date: 10 Dec 2009).

[76] Coenraets, C. Collaborative data entry with Flex and BlazeDS (online), http://www.adobe.com/devnet/flex/articles/data_entry.html (Access date: 5 May 2008).

[77] Wikipedia List of social networking websites (online), http://en.wikipedia.org/wiki/List_of_social_networking_websites (Access date: 28 Oct 2009).

[78] facebook_developers_blog The Facebook Open Stream API (online), http://developers.facebook.com/news.php?blog=1&story=225 (Access date: 29 Oct 2009).

[79] Dalrymple, J. Adobe, Facebook partner to create Flash developer tools (online), http://www.macworld.com/article/139727/2009/03/adobefacebook.html (Access date: 28 Oct 2009).

[80] adobe.com Adobe Developer Connection: Facebook (online), http://www.adobe.com/devnet/facebook/ (Access date: 28 Oct 2009).

[81] twitter.com twitter (online), http://twitter.com/ (Access date: 28 Oct 2009).

[82] Wikipedia Twitter (online), http://en.wikipedia.org/wiki/Twitter (Access date: 28 Oct 2009).

[83] Cheng, A. and Evans, M. Inside Twitter: An In-Depth Look Inside the Twitter World (online), Sysomos Inc., http://sysomos.com/insidetwitter/ (Access date: 28 Oct 2009).

[84] tweetdeck.com TweetDeck (online), http://tweetdeck.com/beta/ (Access date: 28 Oct 2009).

[85] Coenraets, C. Google Map Collaboration (online), http://coenraets.org/blog/2008/05/google-maps-collaboration-using-googles-new-actionscript-api-and-blazeds/ (Access date: 20 July 2009).

[86] google.com Google Wave (online), http://wave.google.com/help/wave/about.html (Access date: 24 Oct 2009).

[87] maximumpc.com Google Wave to Open its Own App Store and Itself to Software Developers (online), http://www.maximumpc.com/article/news/google_wave_open_its_own_app_store_and_itself_software_developers (Access date: 29 Oct 2009).

[88] ribbit.com Ribbit (online), http://www.ribbit.com/ (Access date: 24 Oct 2009).

[89] ribbit.com Real-Time Conversation Streams in Google Wave Powered by Ribbit (online), http://www.ribbit.com/wave/ (Access date: 29 Oct 2009).

[90] rstoeber.com Google Voice Desktop App (online), http://rstoeber.com/apps/Google_Voice_Utility.html (Access date: 21 Oct 2009).

[91] Scaleform Scaleform GFx (online), http://www.scaleform.com/ (Access date: 12 Jan 2010).

[92] Augustyn, T. Realtime Terminator Salvation "Machine Vision" fx (online), http://play.blog2t.net/terminator-salvation-realtime-machine-vision-as3/#machine_vision_demo (Access date: 19 Jan 2010).

[93] Krause, J. ThunderBolt AS3: Logger Extension for Flex, AIR, and Flash Applications (online), http://code.google.com/p/flash-thunderbolt/ (Access date: 20 Nov 2009).

[94] McNeely, M. Google Analytics within Flex/Flash Applications (online), http://www.insideria.com/2009/02/using-google-analytics-within.html (Access date: 20 Nov 2009).

[95] Reimers, S. and Stewart, N. (2007), Adobe Flash as a medium for online experimentation: A test of reaction time measurement capabilities, *Behavior Research Methods*, 39 (3), 365-370.

[96] Millet, B., Asfour, S. and Lewis, J.R. (2009), Selection-based virtual keyboard prototypes and data collection application, *Behavior Research Methods*, 41 (3), 951-956.

[97] Weber, S. Free Flash Scrollbar with Throwing Physics (online), http://www.weberdesignlabs.com/blog/2008/08/free-flash-scrollbar-with-throwing-physics/ (Access date: 29 May 2009).

[98] jiglibflash jiglibflash:AS3 3D Phsics Engine (online), http://www.jiglibflash.com/blog/ (Access date: 1 Oct 2009).

[99] WOW-Engine WOW-Engine: AS3 3D Physics Engine (online), http://code.google.com/p/wow-engine/ (Access date: 1 Oct 2009).

[100] Fisix_Engine The Fisix Engine: AS3 Physics Engine for Game Developers (online), http://www.fisixengine.com/default.asp (Access date: 1 Oct 2009).

[101] axiis Axiis Flex Data Visualization API (online), http://www.axiis.org (Access date: 28 Sep 2009).

[102] Gonzalez, T. Oceanography Visualization (online), http://www.twgonzalez.com/blog/?p=280 (Access date: 22 Oct 2009).

[103] SpatialKey Harnessing the power of City data with SpatialKey (online), http://blog.spatialkey.com/2009/10/city-data-with-spatialkey/ (Access date: 20 Oct 2009).

[104] google.com Google Finance: Real Estate Index (online), http://www.google.com/finance?q=GOOGLEINDEX_US:RLEST (Access date: 30 Oct 2009).

[105] Raad, M. Spatial / Temporal MBTA HeatMap (online), http://thunderheadxpler.blogspot.com/2009/11/spatial-temporal-mbta-heatmap.html (Access date: 14 Nov 2009).

[106] flare.prefuse.org flare: Data Visualization for the Web (online), http://flare.prefuse.org/ (Access date: 12 May 2008).

[107] Google Google Maps API for Flash (online), http://code.google.com/apis/maps/documentation/flash/ (Access date: 24 Sep 2009).

[108] Adobe.com Flash Player 10 Features: Hardware Acceleration (online), http://www.adobe.com/products/flashplayer/features/ (Access date: 28 Sep 2009).

[109] Uro, T. What Does GPU Acceleration Mean? (online), http://www.kaourantin.net/2008/05/what-does-gpu-acceleration-mean.html (Access date: 28 Sep 2009).

[110] Wikipedia 3D Flash (online), http://en.wikipedia.org/wiki/3D_Flash (Access date: 28 Sep 2009).

[111] UK_Army Start Thinking Soldier (online), https://www.armyjobs.mod.uk/startthinkingsoldier/Pages/Default.aspx (Access date: 24 Sep 2009).

[112] DesignReviver 15 Amazing 3D Flash Websites (online), http://designreviver.com/inspiration/15-amazing-3d-flash-websites/ (Access date: 24 Sep 2009).

[113] McCann_Erickson Eco Zoo Website (online), http://ecodazoo.com/ (Access date: 24 Sep 2009).

[114] nationalgeographic.com The Globe of Human History (online), https://genographic.nationalgeographic.com/genographic/lan/en/globe.html (Access date: 2 dec 2009).

[115] Santander, A. Warping 3D Text (online), http://away3d.com/warping-3d-text (Access date: 28 Sep 2009).

[116] Henderson, S. and Feiner, S. Augmented Reality for Maintenance and Repair (ARMAR) (online), http://graphics.cs.columbia.edu/projects/armar/index.htm (Access date: 27 Oct 2009).

[117] vesseltracker.com Vesseltracker Layar (online), http://www.vesseltracker.com/en/static/Mobile.html (Access date: 19 Nov 2009).

[118] General_Electric_Company Smart Grid Augmented Reality (online), http://ge.ecomagination.com/smartgrid/#/augmented_reality (Access date: 27 Oct 2009).

[119] US_Postal_Service Virtual Box Simulator (online), https://www.prioritymail.com/simulator.asp (Access date: 28 Oct 2009).

[120] Spark_Project Flash Augmented Reality Tool Kit - FLARToolKit (online), http://www.libspark.org/wiki/saqoosha/FLARToolKit/en (Access date: 10 Sep 2009).

[121] Brimelow, L. Introduction to Augmented Reality (online), http://www.gotoandlearn.com/play?id=105 (Access date: 10 Sep 2009).

[122] Adobe.com Augmented Reality Using a Webcam and Flash (online), http://www.adobe.com/devnet/flash/articles/augmented_reality.html (Access date: 10 Sep 2009).

[123] PaperVision3D PaperVision3D Project Home (online), http://code.google.com/p/papervision3d/ (Access date: 10 Sep 2009).

[124] FLARToolKit FLARToolKit User Group on Google Code (online), http://groups.google.com/group/flartoolkit-userz/web/links (Access date: 10 Sep 2009).

[125] Ideum Ideum Company Website (online), http://www.ideum.com/ (Access date: 22 Sep 2009).

[126] GestureWorks GestureWorks: Flash multi-touch framework and SDK (online), http://gestureworks.com/ (Access date: 22 Sep 2009).

[127] Adobe.com Inspire: Adobe and the Future of Multi-touch (online), https://xd.adobe.com/#/featured/video/160 (Access date: 22 Sep 2009).

[128] labs.ideo.com ideo-multitouch: Multitouch Package for Flash & Processing (online), http://code.google.com/p/ideo-multitouch/ (Access date: 21 Sep 2009).

[129] NUI_Group Building Your First Multi-Touch Application in Flash (online), http://wiki.nuigroup.com/Building_Your_First_Multi-Touch_Application_in_Flash (Access date: 15 Sep 2009).

[130] Zhang, M. The State of Flex RIA Development Ecosystem (online), http://www.infoq.com/news/2010/01/state-of-flex-dev-tools (Access date: 11 Jan 2010).

[131] adobe.com Flex version comparison chart (online), http://www.adobe.com/products/flex/upgrade/ (Access date: 17 Dec 2009).

[132] Grigg, D. When to Flash and when to Flex (online), http://www.dgrigg.com/post.cfm/11/04/2009/When-to-Flash-and-when-to-Flex (Access date: 5 Nov 2009).

[133] Shorten, A. (2009), Design to Development: Flash Catalyst to Flash Builder, In *Proceedings of Adobe Max 2009*, Los Angeles, CA.

[134] adobe.com Adobe Flash Catalyst (online), http://labs.adobe.com/technologies/flashcatalyst/ (Access date: 30 Oct 2009).

[135] Shankland, S. Adobe wants to bridge gap between PCs and cloud (online), cnet, http://news.cnet.com/8301-1001_3-10097640-92.html (Access date: 17 Dec 2009).

[136] wikipedia.org Adobe LiveCycle (online), Wikipedia, The Free Encyclopedia. , http://en.wikipedia.org/wiki/Adobe_LiveCycle (Access date: 18 Dec 2009).

[137] Thompson, B. (2009), Creating Portable RIAs, In *Proceedings of Adobe Max 2009*, Los Angeles, CA.

[138] Adobe Adobe® LiveCycle® solutions for intuitive user experiences (online), http://www.adobe.com/products/livecycle/pdfs/bpm_wp_final.pdf (Access.

[139] Powerflasher FDT (online), http://fdt.powerflasher.de/developer-tools/fdt-3/home/ (Access date: 17 Sep 2009).

[140] Keefe, M. Scriptplayground Reviews: FDT 3.0 Editor (online), http://scriptplayground.com/reviews/fdt_3_0.php (Access date: 17 Sep 2009).

[141] Brimelow, L. The Flash Blog: ActionScript Editor (online), http://theflashblog.com/?cat=45 (Access date: 17 Sep 2009).

[142] wonderfl wonderfl: build flash online (online), http://wonderfl.net/ (Access date: 17 Sep 2009).

[143] TechCrunch Sprout: The Online WYSIWYG Editor for Flash (online), http://www.techcrunch.com/2008/01/29/sprout-the-online-wysiwyg-editor-for-flash/ (Access date: 24 Oct 2009).

[144] Garvey, R. Online Dynamic Content Builder (online), http://www.ndlr.ie/telcop/?p=31 (Access date: 12 Jan 2010).

[145] sapphiresteel.com Amethyst: Visual Studio IDE for the Flash Platform (online), http://www.sapphiresteel.com/ (Access date: 30 Oct 2009).

[146] ensemble.com Tofino (online), http://www.ensemble.com/products/tofino.shtml (Access date: 21 Oct 2009).

[147] FlashDevelop FlashDevelop Web Site (online), http://www.flashdevelop.org/wikidocs/index.php?title=Main_Page (Access date: 17 Sep 2009).

[148] Low, L. Review: FDT Vs FlashDevelop (online), http://blog.flashmech.net/2008/08/review-fdt-vs-flashdevelop/ (Access date: 17 Sep 2009).

[149] MiniBuilder MiniBuilder: Flash based IDE for ActionScript (online), http://code.google.com/p/minibuilder/ (Access date: 28 Oct 2009).

[150] Labs, A. (16 Jan 2009), Alchemy:FAQ (online), http://labs.adobe.com/wiki/index.php/Alchemy:FAQ (Access date: 8 Jul 2009). Alchemy

[151] adobe.com Flex Quick Start: Building custom components (online), http://www.adobe.com/devnet/flex/quickstart/deploying_components/ (Access date: 30 Oct 2009).

[152] adobe.com Adobe Exchange (online), http://www.adobe.com/exchange/ (Access date: 11 Dec 2008).

[153] adobe.com Adobe Extension Manager (online), http://www.adobe.com/exchange/em_download/ (Access date: 23 May 2008).

[154] afcomponents.com Advanced Flash Components (online), http://www.afcomponents.com/ (Access date: 30 Apr 2008).

[155] flex.org Flex.org: Components (online), http://flex.org/software/components (Access date: 9 July 2008).

[156] riaforge.org RIAForge (online), http://www.riaforge.org/ (Access date: 30 Oct 2009).

[157] activeden.net ActiveDen (online), http://activeden.net/ (Access date: 30 Oct 2009).

[158] flexden.net FlexDen (online), http://www.flexden.net/ (Access date: 30 Oct 2009).

[159] flashloaded.com Flashloaded (online), http://www.flashloaded.com/ (Access date: 30 Oct 2009).

[160] Kromkamp, B. RIA frameworks - a comparison (online), http://www.quesucede.com/page/show/id/ria_frameworks (Access date: 17 Dec 2009).

[161] gnu.org gnash (online), gnu, http://www.gnu.org/software/gnash/ (Access date: 18 Dec 2009).

[162] openlaszlo.org OpenLaszlo (online), http://www.openlaszlo.org/ (Access date: 18 Dec 2009).

[163] haxe.org haXe (online), http://haxe.org/ (Access date: 13 Oct 2009).

[164] wikipedia.org Microsoft Silverlight (online), Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/wiki/Microsoft_Silverlight (Access date: 18 Dec 2009).

[165] asual.com SWFAddress: Deep linking for Flash and Ajax (online), http://www.asual.com/swfaddress/ (Access date: 18 Dec 2009).

[166] wikibin.org Criticism of Adobe Flash (online), http://wikibin.org/articles/criticism-of-adobe-flash.html (Access date: 18 Dec 2009).

[167] Farrel, N. Adobe will replace Microsoft as hacker's target (online), Fudzilla, http://www.fudzilla.com/content/view/17019/38/ (Access date: 30 Dec 2009).

[168] adobe (2009), Adobe Max 2009: Keynote Day 1, In *Proceedings of Adobe Max 2009*, Los Angeles, CA.

[169] Husada, R. Flash for iPhone development confirmed (online), http://www.rivalschools.tv/tag/adobe/ (Access date: 21 Jan 2010).

[170] Chandna, P. Mobile Internet to be 2x Desktop Internet, says Morgan Stanley (online), MAXIMUMPC, http://www.maximumpc.com/article/news/mobile_internet_be_2x_desktop_internet_says_morgan_stanley (Access date: 21 Dec 2009).

[171] intel Intel Unveils 45nm System-on-Chip for Internet TV (online), http://www.intel.com/pressroom/archive/releases/20090924comp_b.htm (Access date: 24 sep 2009).

[172] Voerman, M. Adobe AIR – The future platform of digital publishing? (online), http://blog.schematic.com.au/?p=98 (Access date: 9 Dec 2009).

[173] Deveria, A. Compatibility tables for features in HTML5, CSS3, SVG and other upcoming web technologies (online), http://a.deveria.com/caniuse/ (Access date: 20 Nov 2009).

[174] Alexander, V. Is HTML5 good for application developers? (online), http://rebuildingtheweb.com/en/is-html5-good-for-app-developers/ (Access date: 23 Nov 2009).

[175] Kirk, J. Browser vendor squabbles cause W3C to scrap codec requirement (online), http://infoworld.com/d/developer-world/browser-vendor-squabbles-cause-w3c-scrap-codec-requirement-974 (Access date: 23 Nov 2009).

[176] Paul, R. Decoding the HTML 5 video codec debate (online), http://arstechnica.com/open-source/news/2009/07/decoding-the-html-5-video-codec-debate.ars (Access date: 23 Nov 2009).

[177] Lengelé, Q. BumpMapping with Flash (online), http://www.cornflex.org/?p=28 (Access date: 8 Apr 2008).

[178] Doyle, J. TweenLite – A Lightweight, FAST Tweening Engine (online), http://blog.greensock.com/tweenlite/ (Access date: 21 Sep 2009).

[179] O'Neil, C. Collision Detection Kit (online), http://www.coreyoneil.com/portfolio/index.php?project=5 (Access date: 26 Oct 2009).

[180] Pesenti, P. Actionscript 3 - as3 Lightning / Thunderbolt / Electric discharge Class (online), http://blog.oaxoa.com/2009/07/27/actionscript-3-as3-lightning-thunderbolt-electric-discharge-class/ (Access date: 28 July 2009).

[181] alternativaplatform.com Mobile Phone (online), http://alternativaplatform.com/swf/demos/mobilephone/mobilephone.swf (Access date: 13 Dec 2008).

[182] Gurgul, Y. Throwing Dice With the Jiglib Physics Engine and Away3D (online), http://www.flashperfection.com/tutorials/Throwing-Dice-With-the-Jiglib-Physics-Engine-and-Away3D-00303.html (Access date: 5 Oct 2009).

[183] Away3D Away3D Web Site (online), http://away3d.com/ (Access date: 24 Sep 2009).

[184] adobe.com Media sample: Sound channels (online), http://www.adobe.com/devnet/flash/samples/media_5/index.html (Access date: 03 May 2009).

[185] Meutzner, B. Flex Chart Range Selector - Google Finance'ish (online), http://www.stretchmedia.ca/blog/index.cfm/2007/3/9/Flex-Chart-Range-Selector--Google-Financeish (Access date: 12 Mar 2009).

[186] box2dflash.sourcefourge.net Box2DFlashAS3 2.0.1 (online), http://box2dflash.sourceforge.net/ (Access date: 29 Apr 2008).

This page intentionally left blank.

Annex A Embedded Flash Samples

This Annex contains embedded Flash content which requires Adobe Reader version 9.0 or newer to view. Simply click the content to activate it. Once activated, content can be disabled at any time by right-clicking and selecting 'Disable Content' from the context menu. For simplicity and security reasons, all of the Flash applications are strictly client based; none of the included examples require a network connection. Note performance[6] will vary sharply depending on the user's local hardware capabilities. Several of the demos are experimental in nature and as such extremely performance demanding. It is therefore strongly recommended readers activate just one demo at a time.

[6] All of the embedded examples were compiled for the older version 9 of the Flash player (for compatibility with Acrobat 9), and as such may not represent performance with the more optimized current (version 10) Flash player.

A.1.1 Graphics – Drawing

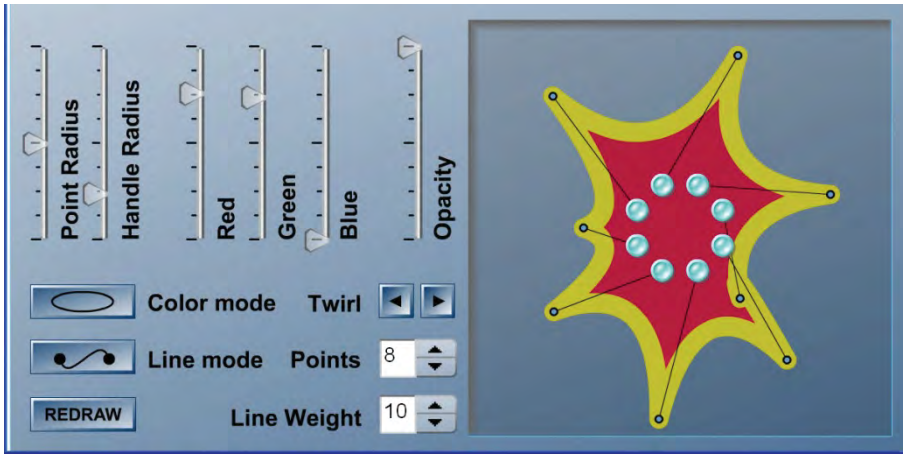

Figure A-103: Interactive sample of Flash player's (runtime) drawing features[8]. After activating the demo, use the controls to interactively modify the graphic.

A.1.2 Graphics – Image Masking

Figure A-104: Simple example of image masking in Flash[7]. After activating the demo, click the red buttons to toggle between masking examples.

A.1.3 Interactive Graphics – Bump Mapping with Dynamic Lighting

Figure A-105: An interactive example of bump mapping[177] with dynamic lighting created using ActionScript. Use the mouse to adjust the light source (mouse move = position, mouse click = color, mouse wheel = specularity, and ctrl+ mouse wheel = spot size). The space bar can be used to toggle surface texture.

A.1.4 Graphic Animation – Motion and Property Tweening

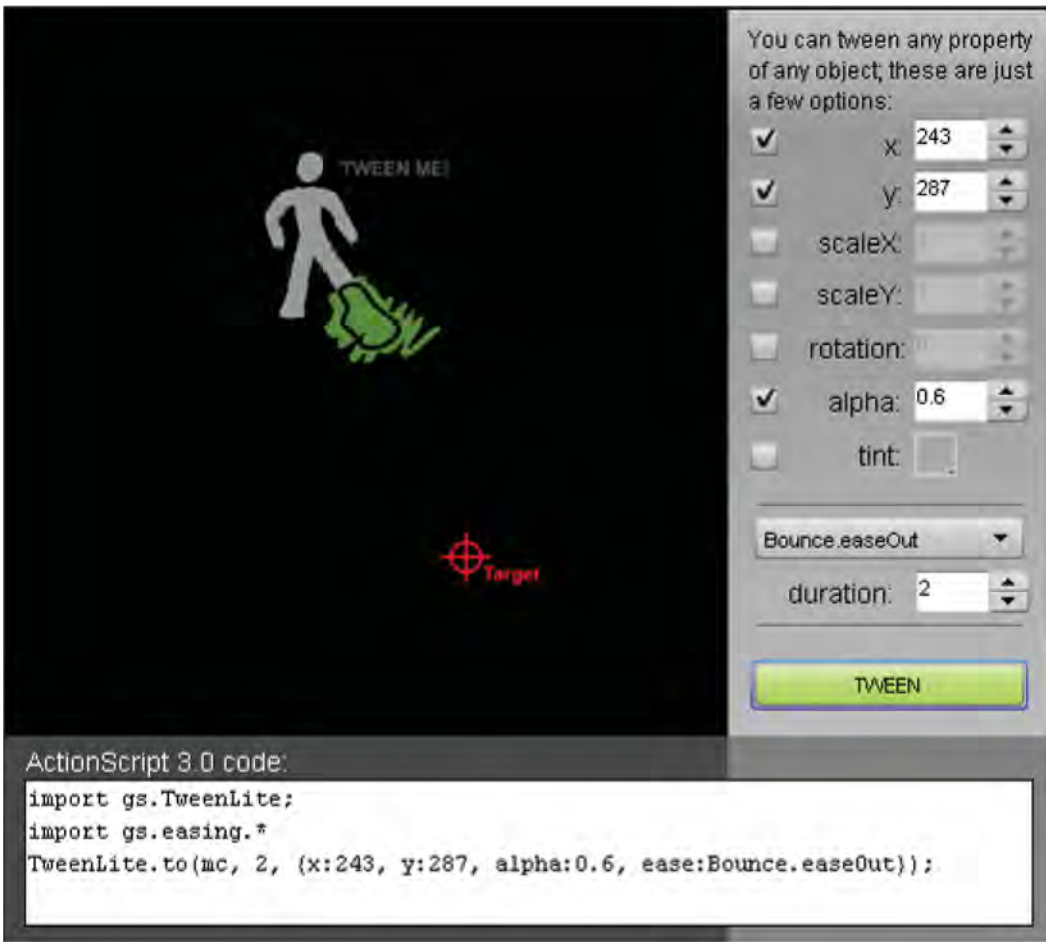

Figure A-106: Interactive demo of Flash animation using GreenSock's TweenLite[178] API. After activating the demo, drag the red crosshairs to any desired position, select an easing function, and click the 'TWEEN' button. Use the controls provided to select and adjust the animation properties. The ActionScript 3 code used to implement each tween animation is displayed in the bottom text panel for reference.

A.1.5 Collision Detection - Runtime Drawing

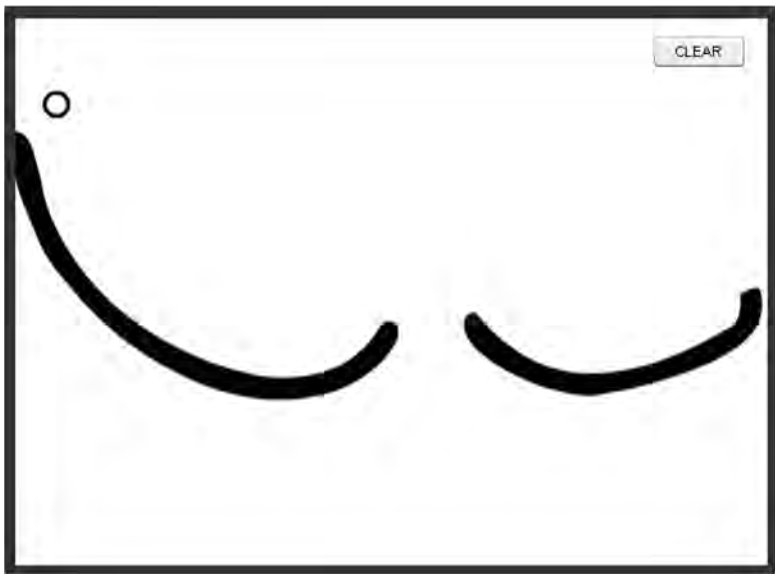

Figure A-107: A demo of runtime drawing and graphic collision detection in Flash using the Collision Detection Kit (CDK)[179]. After activating the demo, a ball will fall until it collides with any drawing lines that are present. Use the 'Clear' button to reset to a blank screen. Create new lines by dragging with the mouse; press and hold the left mouse button while moving the mouse around.

A.1.6 Collision Detection - Text

Figure A-108: Interactive demo of dynamic text and graphic collision detection in Flash using the CDK[179]. After activating the demo, click the text 'EDIT ME!' to edit text as desired. The falling balls will collide with any visible text.

A.1.7 Dynamic Graphic Effects - Lightning

Figure A-109: Interactive demo of Flash dynamic graphic effects[180]. After activating the demo, drag the crosshairs to manipulate the lightning. Use the provided controls to dynamically adjust the visual characteristics and behaviour of the lightning.

A.1.8 3D Graphics – Mobile Phone

Figure A-110: Interactive demo of 3D graphics in Flash using the Altenativa API[181]. After activating the demo, rotate the mobile phone in 3D space using keyboard arrow keys or dragging with the mouse.

A.1.9 3D Physics - Dice

Figure A-111: Interactive dice rolling demo[182] uses both the Away3D[183] graphics API and the Jiglib[98] physics engine. After activating the demo, simply move mouse to position the board and click to throw the dice.

A.1.10 Interactive Audio Mixer

Figure A-112: The audio mixer sample application "demonstrates several sound channels playing together in a mixing board"[184]. After activating the demo, use the stop and play buttons (lower left) to begin tracks, use the sliders to adjust the gain of individual tracks, and the dials to pan from left to right channels.

A.1.11 Interactivity – Mouse Gestures

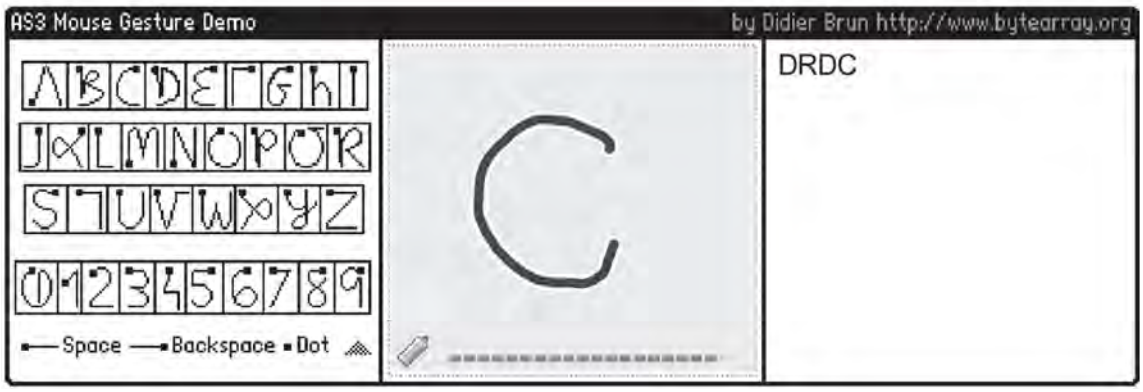

Figure A-113: The Mouse Gesture Demo[42] recognizes specific mouse movement sequences and translates them into text characters. Click and drag the mouse pointer in the grey center panel to create characters as per the legend in the left panel. The recognized gesture is output as a corresponding character in the right panel.

A.1.12 Interactive Visualization – Range Selector Line Chart

Figure A-114: A simple example of an interactive line chart visualization[185]. The histogram doubles as a time range selector.

A.1.13 Tactical Interface Design

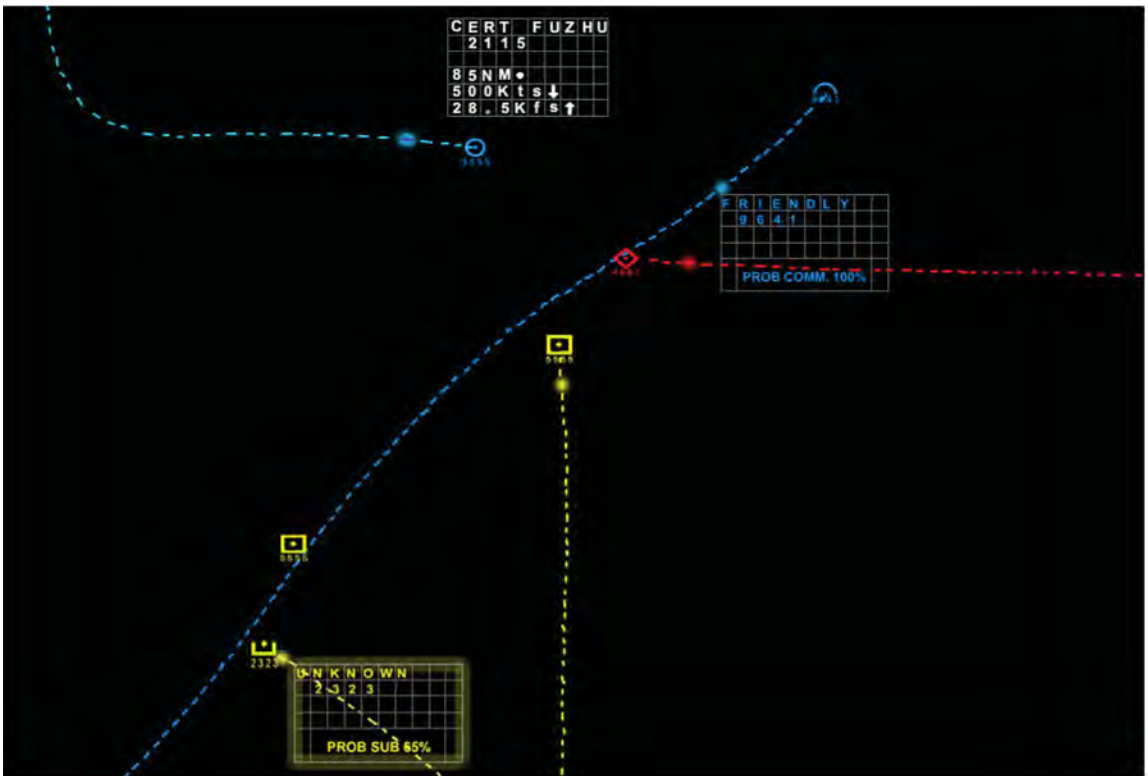

Figure A-115: The Track Data Block demo[29] uses Flash animation and interactivity to experiment with tactical interface design concepts. After activating the demo, use the mouse to hover over targets and invoke differing tooltip styles. Use the friendly surface target (blue circle) to manually scrub playback.

A.1.14 Visualization Samples Using Flare API

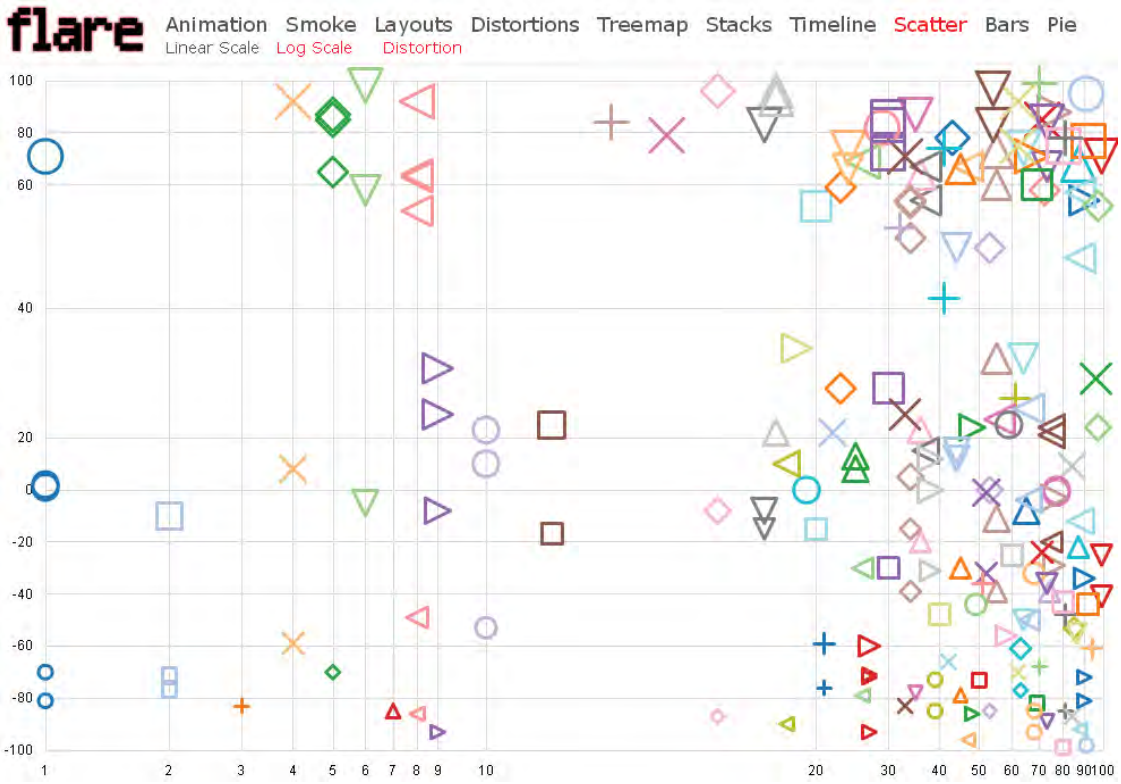

Figure A-116: Interactive and dynamic visualization samples using the flare[106] data visualization API. After activating the demo, use the menu at the top of the screen to navigate to different visualization examples. Use the mouse to experiment with visualizations that support interactivity.

A.1.15 Rich Interactive Application – Mortgage Calculator

Figure A-117: The Mortgage Calculator[59] is a simple example of a small client-side RIA. After activating the demo, use either the input fields directly or their sliders to enter mortgage values. Click 'Calculate' to compute payment info and mouse over the bar graph for yearly details.

A.1.16 Rich Interactive Application - Logicly

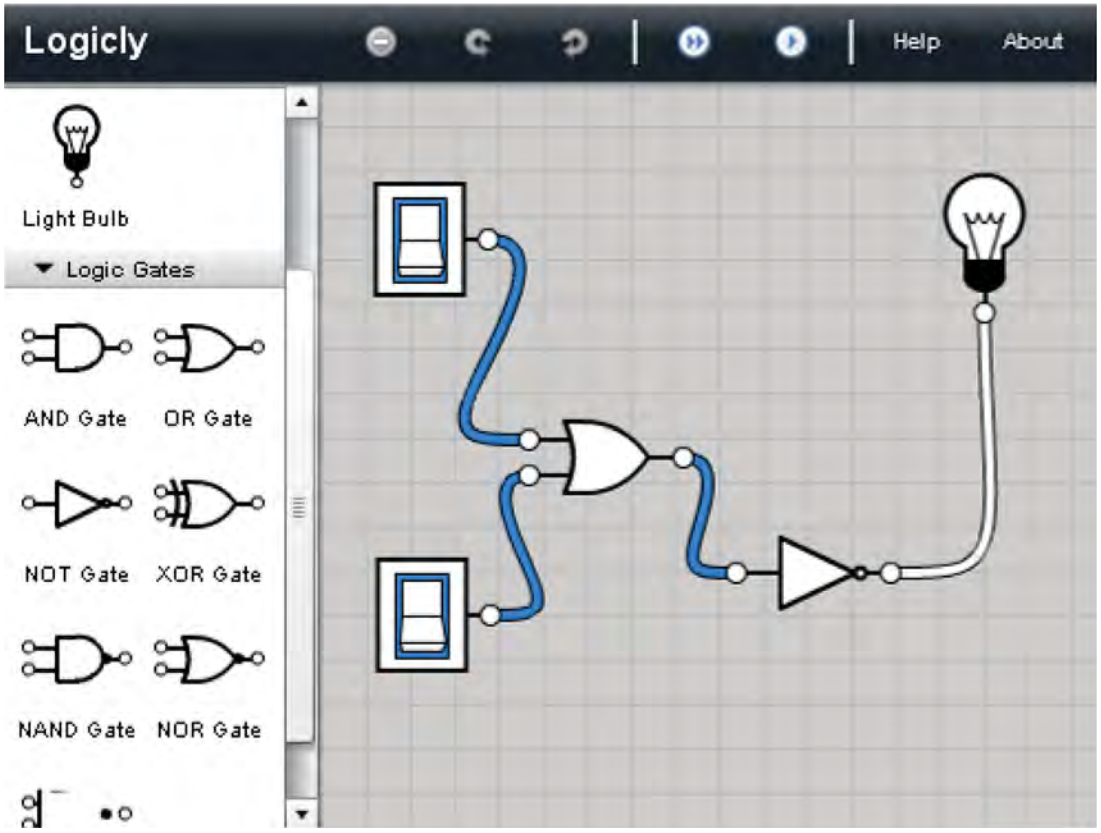

Figure A-118: Logicly[9] is a logic gate simulator and a simple example of a web application with a rich user interaction. After activating the demo, drag and drop circuitry components to the main staging area and connect them using the mouse. Click switches to toggle their state.

A.1.17 Interactive 2D Physics

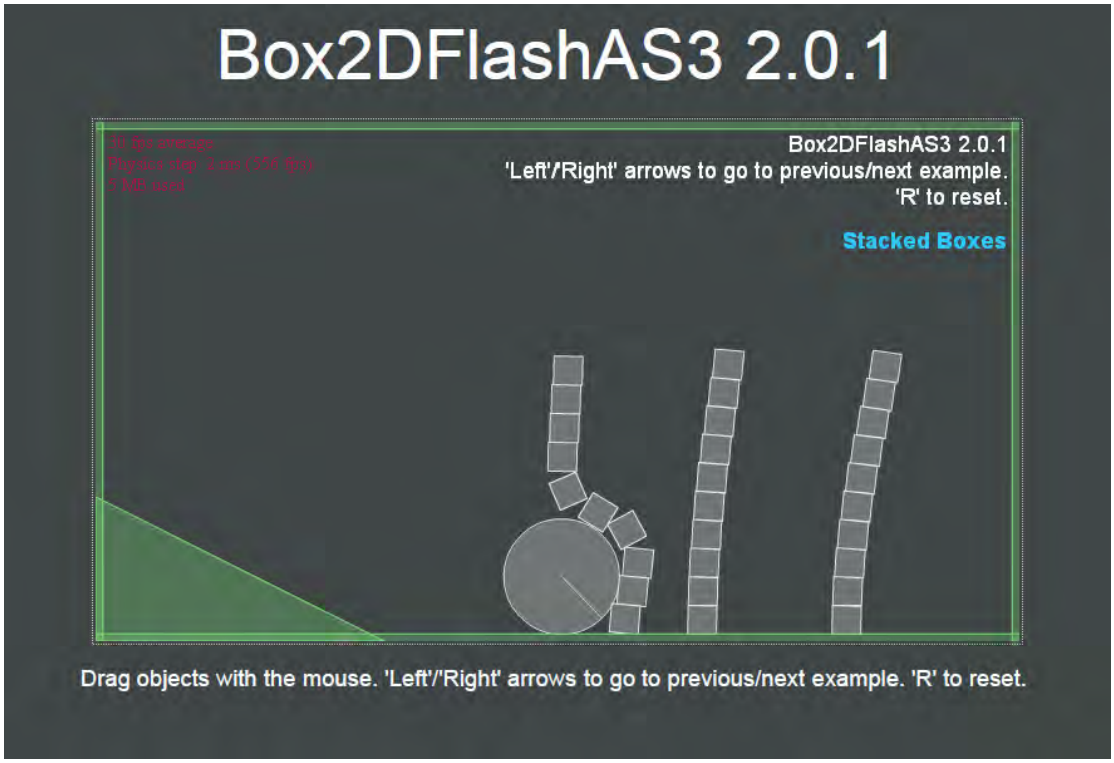

Figure A-119: Interactive demo of 2D physics in Flash using the Box2DFlashAS3[186] open source API. After activating the demo, use the left/right arrow keys to change demos. Use the mouse dragging to move and throw objects. Use 'r' to reset.

A.1.18 Geo-Spatial Temporal Data Visualization

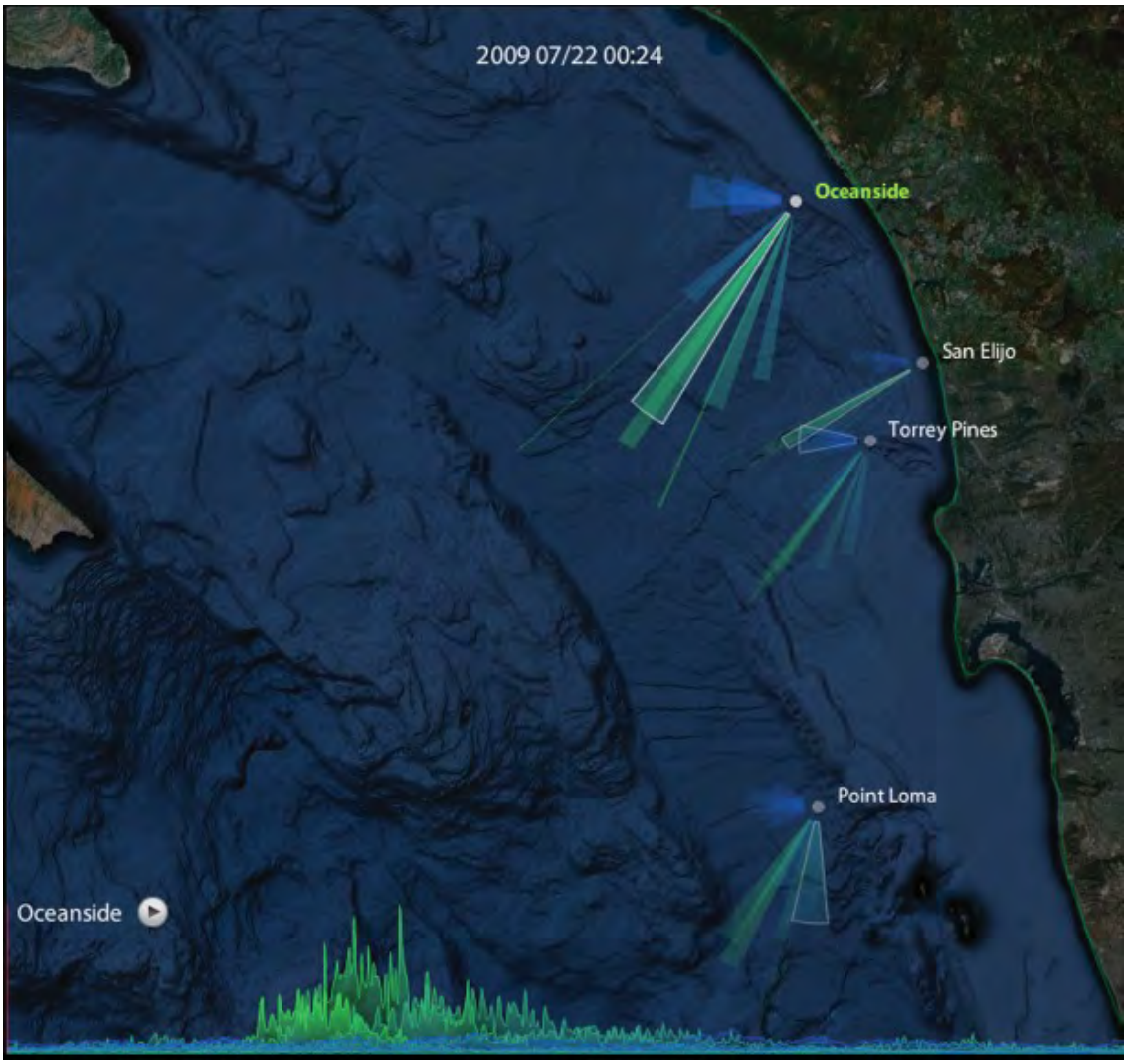

Figure A-120: Oceanography Visualization is an interactive animated visualization of the frequency, direction, and strength of wind waves (blue) and ground swell waves (green) over time. A histogram of wave strength provides temporal navigation in addition to longer term trend patterns. " Interactivity:

1) *You can select a particular buoy by clicking its label - that will then display a 2 week history for that buoy in the histogram along the bottom.*
2) *You can roll over any particular wedge and see a datatip with a 12 hour history for that particular period. The energy shown is in cm2 and represents the surface area underneath the wave - it does not directly correspond to wave height.*
3) *You can animate the 2 week period by pressing the play button within the histogram.*
3) *You can pause the animation and manually move the animation forward with left/right arrow keys*
4) *You can also use your mouse click to skip to any section of the time period."[102]*

This page intentionally left blank.

List of symbols/abbreviations/acronyms/initialisms

AIR	Adobe Integrated Runtime
AJAX	Asynchronous JavaScript And XML
API	Application Programming Interface
AR	Augmented Reality
AS3	ActionScript 3
AVM	ActionScript Virtual Machine
BCI	Brain-Computer Interface
CPU	Central Processing Unit
CS3/CS4	Creative Suite 3/Creative Suite 4
DND	Department of National Defence
DRDC	Defence Research & Development Canada
DRDKIM	Director Research and Development Knowledge and Information Management
FLV	Flash Video
FMS	Flash Media Server
FPS	First Person Shooter or Frames Per Second (depending on context)
GIS	Geographic Information System
GPS	Global Positioning System
GPU	Graphical Processing Unit
GUI	Graphical User Interface
HCI	Human Computer Interface
HITL	Human-In-The-Loop
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
HUD	Heads-Up Display
IaaS	Infrastructure as a Service
IDE	Integrated Development Environment
IP	Internet Protocol
ISO	International Standards Organization
JAR	Java ARchive
JIT	Just-in-Time

JRE	Java Runtime Engine
LCDS	LiveCycle Data Services
MC	Movie Clip
MC2CD	Maritime Command and Control Concept Development
MICS	Maritime Information and Combat Systems
MMO	Massive Multi-player Online
MXML	Macromedia eXtensible Markup Language
MXP	Macromedia eXtension Package
NUI	Natural User Interface
OOB	Out-of-Box
OOP	Object-Oriented Programming
OS	Operating System
OSC	Open Sound Control
PaaS	Platform as a Service
PBJ	Pixel Bender JIT (Just In-Time) compiler
PBK	Pixel Bender Kernel
PDF	Portable Document Format
RIA	Rich Internet Application
RPG	Role-Playing Game
R&D	Research & Development
SaaS	Software as a Service
SDK	Software Development Kit
SME	Subject Matter expert
SMS	Short Message Service
SQL	Structured Query Language
SWF	Shockwave Flash
TCP	Transfer Control Protocol
TLF	Text Layout Framework
TUIO	Tangible User Interface OSC (Open Sound Control)
UDP	Universal Data Packet
UI	User Interface
URL	Universal Resource Locator

VIPER	Virginia Interoperability Picture for Emergency Response
VOIP	Voice Over Internet Protocol
WOCA	Write-Once-Compile-Anywhere
WORA	Write-Once-Run-Anywhere
WWW	World Wide Web
WYSIWYG	What-You-See-Is-What-You-Get
W3C	World Wide Web Consortium
XML	eXtensible Markup Language

This page intentionally left blank.

Distribution list

Document No.: DRDC Atlantic TM 2008-291

LIST PART 1: Internal Distribution by Centre

1 Don Coady
1 Mark Hazen
1 Francine Desharnais
1 Bruce Chalmers
1 Glenn Franck
1 Liesa Lapinski
1 Andrew MacInnis
1 Jacqui Crebolder
1 Anthony Isenor
1 Brad Dillman
3 DRDC Atlantic Library (1 hard copy, 2 CDs)

13 TOTAL LIST PART 1

LIST PART 2: External Distribution by DRDKIM

1 Library and Archives Canada
1 DRDKIM

DRDC Toronto

1 Allan Keefe
1 Dave Eaton
1 Adele Mininni
1 Maj Stephen Boyne
1 Maj Linda Bossi
1 Philip Terhaar
1 Renee Chow
1 Sharon McFadden
1 Peter Kwantes

11 TOTAL LIST PART 2

24 TOTAL COPIES REQUIRED

This page intentionally left blank.

DOCUMENT CONTROL DATA

(Security classification of title, body of abstract and indexing annotation must be entered when the overall document is classified)

1. ORIGINATOR (The name and address of the organization preparing the document. Organizations for whom the document was prepared, e.g., Centre sponsoring a contractor's report, or tasking agency, are entered in section 8.)

Defence R&D Canada – Atlantic
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7

2. SECURITY CLASSIFICATION (Overall security classification of the document including special warning terms if applicable.)

UNCLASSIFIED

3. TITLE (The complete document title as indicated on the title page. Its classification should be indicated by the appropriate abbreviation (S, C or U) in parentheses after the title.)

Flash Platform Examination

4. AUTHORS (last name, followed by initials – ranks, titles, etc. not to be used)

Coady, D.

5. DATE OF PUBLICATION (Month and year of publication of document.)

March 2011

6a. NO. OF PAGES (Total containing information, including Annexes, Appendices, etc.)

188

6b. NO. OF REFS (Total cited in document.)

186

7. DESCRIPTIVE NOTES (The category of the document, e.g., technical report, technical note or memorandum. If appropriate, enter the type of report, e.g., interim, progress, summary, annual or final. Give the inclusive dates when a specific reporting period is covered.)

Technical Memorandum

8. SPONSORING ACTIVITY (The name of the department project office or laboratory sponsoring the research and development – include address.)

Defence R&D Canada – Atlantic
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)

9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)

10a. ORIGINATOR'S DOCUMENT NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)

DRDC Atlantic TM 2008-291

10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)

11. DOCUMENT AVAILABILITY (Any limitations on further dissemination of the document, other than those imposed by security classification.)

Unlimited

12. DOCUMENT ANNOUNCEMENT (Any limitation to the bibliographic announcement of this document. This will normally correspond to the Document Availability (11). However, where further distribution (beyond the audience specified in (11) is possible, a wider announcement audience may be selected.))

Unlimited

13. ABSTRACT

The Flash platform is a comprehensive infrastructure of runtime clients, developer tools, and server-side technologies for designing, developing, and delivering an extensive variety of software content and applications. Crowning this platform is the world's most ubiquitous application runtime environment - the Flash player, providing developers and end users alike the benefits of a consistent and unifying architecture across competing browsers, desktops, and mobile devices. The core strengths of the platform include multimedia, graphics, animation, and advanced text, all of which can be enhanced with sophisticated interactivity. Arguably more significant is Flash's capability related to developing and deploying second generation web-based programs with desktop style interaction, collectively known as rich internet applications.

As a research tool, Flash offers a highly versatile prototyping and rapid application development environment for exploring, refining, and experimenting with novel user interface concepts. Flash applications of varying fidelity can incorporate advanced data visualizations, novel interactivity, rich animation, and numerous forms of multimedia. Being a predominant and forefront technology of the web, Flash is ideally positioned for leveraging the exploding wealth of cloud based informational services and resources available across the internet.

This document examines the Flash platform from both an end-user and developer perspective, primarily within the context of a research tool for design, development, and experimentation involving defence and security information systems.

La plateforme Flash est une infrastructure complète de clients d'exécution, des outils du développeur et des technologies côté serveur pour la conception, le développement et la prestation d'une vaste gamme de contenu logiciel et des applications. Cette plateforme est l'environnement d'exécution la plus utilisée dans le monde – l'application Flash player, fournit aux développeurs et aux utilisateurs finals les avantages d'une architecture unifiante et constante de tous les navigateurs, ordinateurs de bureau et dispositifs mobiles concurrents. La plateforme présente des forces de base, notamment la capacité multimédia, les graphiques, l'animation et le traitement de texte évolué, qui peuvent être améliorées à l'aide d'une capacité évoluée d'interactivité. La capacité de Flash la plus importante est sans doute sa capacité à développer et à déployer des programmes Web de seconde génération avec une interaction de type bureau, connue sous le nom d'application Internet riche (RIA).

En tant qu'outil de recherche, Flash offre un environnement très souple de développement d'application rapide et de prototypage pour explorer, améliorer et expérimenter les concepts de l'interface utilisateur novateurs. Les applications Flash de diverses fidélités peut intégrer des fonctions de visualisation des données, d'interactivité nouvelle, d'animation riche et diverses formes de multimédia. Étant une technologie du Web prédominante et de premier plan, Flash est positionné de façon idéale pour tirer parti de la richesse explosive des services et ressources informationnels de l'informatique en nuage disponibles sur l'Internet.

Ce document examine la plateforme Flash du point de vue de l'utilisateur final et du développeur, principalement dans le contexte d'un outil de recherche pour la conception, le développement et l'expérimentation impliquant les systèmes de défense et de sécurité de l'information.

14. KEYWORDS, DESCRIPTORS or IDENTIFIERS

Flash; Flex; AIR; Adobe; RIA; interface; software; development; animation; video; internet; pdf; GUI; plug-in; browser GUI; plug-in; browser; web; graphics; design; multimedia

This page intentionally left blank.

Defence R&D Canada

Canada's leader in defence
and National Security
Science and Technology

R & D pour la défense Canada

Chef de file au Canada en matière
de science et de technologie pour
la défense et la sécurité nationale

www.drdc-rddc.gc.ca